VF-01-02-A3

OPERA-2D
USER GUIDE

Vector Fields Limited
24 Bankside
Kidlington
Oxford OX5 1JE
England

This document was prepared using Adobe® FrameMaker®.

CONTENTS

Chapter 1
Structure of the User Guide

Road Map 1-1

Chapter 2
Implementation Notes

UNIX Implementation 2-1
Windows Implementation 2-8

Chapter 3
Program Philosophy

Introduction 3-1
Analysis Programs 3-3
Program Philosophy 3-5

Chapter 4
Getting Started

Starting the OPERA-2d Pre and Post Processor 4-1
Pre Processing 4-2
Solving the Problem 4-34
Post Processing 4-37

Chapter 5
Analysis and Utility Programs

Introduction 5-1
The Finite Element Method 5-2
Application Notes 5-10
The Basic Field Equations used in OPERA-2d 5-13
Static Field Analysis (ST) 5-15
Steady-state AC Analysis (AC) 5-18
Transient Analysis (TR) 5-22
Velocity Analysis (VL) 5-28
Rotating Machines (RM) 5-31
Linear Motion (LM) 5-41
Space Charge Beam Analysis (SP) 5-54
Thermal Analysis (TH and THTR) 5-70
Stress Analysis (SA) 5-74
The DXF Interface Program 5-79
Adaptive Analysis 5-81

Chapter 6
Application Notes

Flux Linkage Calculations in 2D Solutions 6-1
Particle Trajectory Intersections in OPERA-2d 6-4
External Circuits in OPERA-2d 6-5
Accurate Results for Stress Analysis 6-11
Electromagnetic and Stress Analysis of an SRM 6-20
Power and Energy Calculation in AC Solutions 6-28
Inductance Calculations in OPERA-2d 6-30
Use of Command Scripts to Calculate Fourier Series 6-33
Complex Material Properties 6-35
Modelling of Asynchronous (Induction) Motors 6-38
Current Diffusion From a Superconductor 6-44
Plasma Free Surface Emitter in 2D 6-49

Chapter 7
A Transformer Example

Introduction 7-1
Pre Processing - Building the Model 7-2
OPERA-2d Static Analysis Module 7-22
Post Processing - Examining the Solution 7-23

Chapter 8
A Solenoid Example

Introduction 8-1
OPERA-2d Pre Processor 8-3
Analysis 8-29
OPERA-2d Post Processor 8-30

Chapter 9
A Permanent Magnet Motor Example

Introduction 9-1
Pre Processing 9-3
Analysis 9-44
Post Processing 9-45

Chapter 10
Stress Analysis Notes and Examples

A review of stress analysis 10-1
Example 1 - A Simple Beam 10-13
Example 2 - Coupled Problems 10-20

Chapter 11
Thermal Analysis Notes and Examples

Thermal Analysis 11-1
Example 1 - A Simple Heat Bath 11-3
Example 2 - Thermally Coupled Problems 11-9

Chapter 12
Space Charge Example

Introduction 12-1
OPERA-2d Pre Processor 12-3
Creating the Emitter Data 12-11
Space Charge Analysis 12-13
Post Processing 12-14

Chapter 1
Structure of the User Guide

Road Map

The OPERA-2d User Guide is structured into the following chapters.

Implementation Notes

The OPERA-2d software can be used on both PCs and workstations, using a variety of operating systems Each has different ways in which the software should be installed and run. This chapter outlines the differences between the systems as it applies to running the software.

Program Philosophy

An overview is given about the underlying philosophy of the software - the fact that models are created in a pre processor including material definitions and mesh generation, and the computed results viewed and processed in the post processor. Both of these functions are carried out in a single module, called the pre and post processor, so that whilst viewing results, changes can immediately be made to the model in order to optimise a design.

Getting Started

A large portion of the User Guide is in this section, where a detailed description of how a model is prepared and analysed is given. New users are encouraged to spend some time going through this chapter, as it will answer many questions that can otherwise arise when using the software.

Analysis Programs

A review of the analysis programs available in the OPERA-2d suite is given in this chapter. In addition, some details are given on the finite element method and accuracy of the method, along with detailed descriptions of the techniques used in each solver.

Application Notes

This chapter contains a number of useful techniques that can be used for performing various tasks. If a question arises as to how to use the software in a particular way, this chapter should first be consulted in case an answer is presented.

Tutorials

A series of examples using the software is included. Each attempts to highlight a typical application using various analysis modules.

Chapter 2
Implementation Notes

UNIX Implementation

Environment Variables

Setting a single Command to run OPERA

A variable **`vfdir`** should be set to the installation directory (for example */u/vfopera*). OPERA-2d and OPERA-3d, including all analysis programs are then run as follows:

```
$vfdir/opera/dcl/opera.com $vfdir
```

The directory name, **`$vfdir`** must be given as a parameter as shown above, so that the lower level shell files can be accessed properly. The command should be used in the definition of an alias (C-shell) or should be entered into an executable file somewhere in your search path so that a single word can be typed to start the software.

For example (C-shell):

Create an entry in *$home/.cshrc*:

```
alias opera '$vfdir/opera/dcl/opera.com $vfdir'
```

For example (other shells):

Make sure your home directory is included in **`$path`**. Create an executable file called *$home/opera* containing the one line:

```
$vfdir/opera/dcl/opera.com $vfdir
```

Graphics Variables

The environment variable VFGRAPHICS is used to set the output of the software, where the options are SCREEN, FILE, BOTH or NONE. For normal operation, this should be set to SCREEN. The graphics software creates an initial graphics window. There is a default size built into the programs, but this can be over-ridden by environment variables, VFWINDOWW and VFWINDOWH, which give the width and height of the window in pixels. If one of the environment variables does not exist or has a value out of range, a message giving the valid range of values is printed and the default values are used.

The graphics window is positioned at the top left of the screen. Thus the programs should be started from a terminal window at the bottom of the screen. The size and position of the graphics window can be adjusted using the window manager functions available through the mouse buttons. Subsequent pictures are scaled to fit the new window size.

Text and Background

If the environment variable VFINV is set to INVERT, the initial setting of text and background colours will be black on white instead of the default of white on black.

Starting the Software

This document assumes that the environment variables specific to OPERA have been set as explained in the previous section. In most cases it is advisable to run the programs from a suitable user directory rather than the directory containing the Vector Fields software. Similarly, it is also advisable to run the software as a "user" rather than as "system manager" since this protects against accidental over writing of files.

Hence, as a user from a suitable local directory, OPERA may be launched by entering:–

```
opera
```

If both OPERA-2d and OPERA-3d are installed, then the system prompts for a choice and, for example, OPERA-2d could be selected:–

```
2d or 3d processing or QUIT?
2d
```

(If only one of OPERA-3d or OPERA-2d is installed, then this choice is not given).

This is followed by a list of options relating to processing environments and analysis modules available. For example, the pre and post processor could be selected:–

```
Option:
pp
```

If you have not set the display environment variables in your system (see earlier) then you are requested to select a method of graphics display. In this case select the screen:–

```
Graphics options: SCREen, FILE, BOTH or NONE? >scre
```

A graphics window is then opened (in addition to the text window) and control moves to the menu system. Access to the menu system is from the main menu bar at the top of the graphics window.

The command to create the terminal window can be incorporated into the commands to start the programs.

File Names and Extensions

Files created by the programs have extensions supplied by the programs. The extensions are upper-case for upper-case file names and lower-case for lower-case file names. The file name extensions are as follows. Some file extensions are common between 2d and 3d programs. For example *.bh* files.

.bh	B-H data files created by OPERA-2d/PP
.op2	data files
.mesh	mesh files
.“analysis”	result files from analysis (AC, SA, ST, TH, TR,VL or RM)
.res	log files created by the analysis programs
.dem	design environment data file
.emit	emitter files used for space charge problems
.comi	command files
.tracks	particle tracking files.
.table	table files
.ps	postscript format graphics files
.hgl	hpgl format graphics files
.pic	internal picture file format

Various log files are created during the running of the pre and post processor. These are:

*Opera2d_PP_***n**.*lp*	All user input and program responses are recorded here
*Opera2d_PP_***n**.*log*	User input is recorded here. These can be reused as input to the program by converting to *.comi* files

The log files are stored in a sub-directory of the project folder *opera_logs*. The file history is always retained. Each time the software is started, new *log*, *lp* etc. files are generated. Previous files are not over written. The history is retained by attaching a number, **n** to the files. The value of **n** is chosen to be the lowest available integer not already in use. This may lead to a situation where the lowest value of **n** is not necessarily the oldest file: for example if some older files were deleted, those values of **n** are re-used later.

COMI Files

The *.comi* file is a command file, that can be run using the $ COMINPUT command, or using **`File` → `Commands In`** menu. The file is a text file, that can be created using a standard text editor, or copied from a *.log* file. The *.comi* files can also be used to automatically start up the interactive programs. Each interactive program always reads the appropriate *.comi* file on start up (although they are normally empty, the user can add commands if desired).

opera2.comi - The pre and post processor always reads this file first.
opera2_de.comi - The Design Environment pre and post processor always reads this file first.

BH Files

The directory *$vfdir/bh* contains sample BH data files for use with OPERA-3d and OPERA-2d. The directory also includes an index *bh.index*.

Picture File Software

The directory *$vfdir/picout* contains software for replaying picture (*.*pic*) files created by the pre and post processor. It can be used to redisplay graphics in a picture file on the screen or to convert the file to Postscript or HPGL for printing. It is documented in the Reference Manuals under the DEVICE and DUMP commands.

Adjusting the Program Sizes

If you have a FORTRAN compiler, the sizes of some of the programs are adjustable. The standard upper limits on problem size are defined in the Reference Manual.

To enable users who have access to computers with large amounts of memory to solve larger problems, the ‘main’ routines and all the libraries necessary to re-link the programs have also been supplied.

In the OPERA-2d PP directory there is a file called *pp.inc*. This file contains a parameter statement which sets the maximum size. To adjust the size of a program edit the *pp.inc* file and re-link the program using the shellscript file

$vfdir/install/makeprog $vfdir

which supplies the UNIX make command with the required flags

Each main routine also includes a file called *pp.inc*, which contains a parameter statement which sets the maximum size. To adjust the size of a program, change directory to the ‘home’ directory of the program, edit the include file, *pp.inc* and re-link the program using the shellscript file

$vfdir/install/makeprog $vfdir

which supplies the UNIX make command with the required flags

The libraries of compiled FORTRAN supplied on the CD-ROM were compiled with native FORTRAN compilers, so may not be compatible with other compilers.

If the program sizes are adjusted, this fact should be declared when calling for User Support.

Swap space (Virtual Memory)

It should be noted that larger programs need larger amounts of memory. If physical memory is exceeded, swap space will be used instead. Excessive use of swap space will degrade the relative performance of the programs for large problem sizes.

Running OPERA off-line (Advanced Users)

It is sometimes beneficial to run the complete cycle of pre processing, analysis and post processing from a single command line script. It is particularly useful when large amounts of post processing are required or when a repetitive post processing task needs to be carried out. Users familiar with the Vector Fields' command language can use the off-line facility for modifying .OP2 files, running an analysis, post processing and then repeating the cycle many times, all from one script. (Note that since the analysis can be started from within the pre and post processor, the following can also be achieved with a single .comi file).

An example of a C-shell script with comments is shown below:

```
#!/bin/csh
# An example of 'Off line' running of OPERA-2d
# it is run by typing  'script' which should be set as executable

# When the Pre and post Processor is run it uses commands stored in
# a file opera2.comi

# VFGRAPHICS is set to none, so the graphics screen is disabled.

# The commands are from the 'Command language' as defined in the
# reference manual

# START OF SCRIPT

# Set the environment variable responsible for saying what
# happens to  graphics output - here we turn screen output off

setenv VFGRAPHICS none

setenv VFDIR $vfdir

# Removes old files :
rm example.op2
rm example.res
rm opera_logs/Opera2d*.*

unalias cp

# copy the commands in pre1.comi to opera2.comi, which is
# automatically run when the Pre and Post Processor starts

cp pre1.comi opera2.comi

# pre1.comi is a VF command script which could generate
# a .op2 file which must include analysis options

# run the Pre and post Processor using the answers to questions
# stored in 'runpre'

$vfdir/opera/dcl/opera.com $vfdir < runpre

# runpre contains for example:
```

```
# 2D
# PP
# Q

# solve the problem

$vfdir/opera/2d/st/st example.op2

# copy the post processing commands to opera2.comi which is
# automatically run when the Pre and post Processor
# is started

cp post1.comi opera2.comi

# post11.comi is a VF command script which could, for example
# read in a .st  file, calculate some forces and dump the results
# out to another file.

# run the post processor

$vfdir/opera/dcl/opera.com $vfdir < runpost

# runpost contains for example:
# 2D
# PP
# Q

# remove the opera2.comi file, otherwise it
# would be called when the user launched the software for
# normal operation

rm opera2.comi


# please note that when running in this fashion it is useful
# to place the command '$comi mode=cont' at the top of command
# files to stop the text window pausing when full.
```

Windows Implementation

Licensing

All OPERA software running on a PC requires a dongle security device to be attached to one of the parallel ports on the back of the computer. Dongles are programmed at Vector Fields to allow licensed software to be run. If further licences are later purchased, Vector Fields can supply codes and instructions on how to reprogramme the dongle; a new dongle is not required. Most implementations use the local (white) dongle which must be attached to the machine where the software is to be run. It is also possible to have network (red) dongles where the dongle is attached to a server PC and other PCs on this network can then use OPERA.

In order for the dongle installation to work correctly the following stages must be carried out depending on whether you have a local (white) dongle or a network (red) dongle.

Local Dongle

1. Connect the dongle to the parallel port (connect to either port if there is more than one).
2. Install the latest version of OPERA.
3. When the installation completes, the software will prompt you to install the “dongle device driver”. All users installing the software for the first time should answer “yes”. On Windows NT systems, users must have administrator privilege to perform this function.
4. Utilities are provided on the CD and the OPERA console for checking dongle status, removing the device driver and installing the device driver. The install and remove options can be run on NT only by users with Administrator privilege.

Network Dongle Server PC

1. Connect the dongle to the parallel port of the server PC (connect to either port if there is more than one).
2. Install OPERA if required to run on the server PC. (If installing on the server, answer YES to the question regarding installing the dongle device driver and go to step 4).
3. From the start-up screen of the OPERA CD, under network dongle utilities, select “Install dongle device driver”. On Windows NT systems, users must have administrator privilege to perform this function.

4. From the same screen run “Install nethasp licence manager”. This utility must be run on the server PC where the dongle is attached. This will launch the installation of a Nethasp license manager program. Select “typical installation” and the manager should be activated automatically. Comprehensive on-line help is supplied.

A range of tools is provided to help with configuring the network dongle. These can be installed from the CD (under network dongle utilities).

Network Dongle-Client PC

1. Install the latest version of OPERA if required on the client PC.
2. When the installation completes, the software will prompt you to install the “dongle device driver”. This is not necessary as the drivers reside on the server PC.
3. Start the OPERA console (see below) and select:

Options ↓
Licensing → **Set Dongle Type** → **Network**

Updating Licences

If licenses need to be updated (extra software is purchased for example), this can be carried out by the user, and a new dongle is not required.

1. Start OPERA
2. From the main OPERA Console (see below) select:

Options ↓
Licensing → **Update Dongle Licensing...**

3. Select **Read Codes From File** and select the file containing the new codes. The file should be in plain text format.

The OPERA Console

The OPERA Console is started from the menu bar as follows:

Start → **Programs** → **Vector Fields OPERA** → **OPERA 8.5**

Alternatively the console can be started from the system icon tray.

Click on the blue and white VF icon as shown here:  .

The console is the ‘navigation centre’ for the complete suite of OPERA software. It allows the following operations:

- Setting of 2d and 3d project directories
- Launching the 2d and 3d pre and post processors
- Launching analyses and organising batch runs
- Viewing analysis result log (*.res)* and emitter (*.emit)* files
- Changing the CPU priority for an analysis
- Listing and updating licensing
- Adjusting some windows parameters, including reversing the foreground and background colours, adjusting the Graphics Window size and the Text IO history buffer.

All options will be shown in the console menus, but only licensed modules will be available. Contact Vector Fields for licensing information. Using the mouse right button on the system tray VF icon will bring up a similar range of options to those available on the menu bar.

When starting the 2d or 3d pre/post processors for the first time, a project folder will be requested. This can be changed at any time, but you will not be prompted when you subsequently start the software. This is the default folder in which OPERA will be working.

Running OPERA Pre and Post Processing

OPERA-2d

The OPERA-2d pre and post processor is started from the console menu bar as follows:

`OPERA-2d` ↓
`Pre and Post-Processor`

The project or working folder can be changed from the default:

`OPERA-2d` ↓
`Change Project Folder` → `OPERA-2d ...`

The dialogue box will then allow you to browse your computer’s folder structure or enter a path name directly. A new folder name can be added if required.

Running OPERA Analysis Modules

Interactive Solutions

All the analysis options are now set in the pre processing. It is possible to run interactive analyses directly from the pre processor. To interactively run from the console menu bar, the OPERA-2D or OPERA-3D analysis modules are accessed as follows:

```
OPERA-2d/3d ↓
     Interactive Solution ...
```

The appropriate analysis module and *.op2* or *.op3* file are then selected

Once an analysis is started, a window will appear indicating the various stages of the solution process. When complete a Windows message will appear asking if you wish to close the solver window. This gives you the opportunity to scroll through the solution steps if you wish. Alternatively you can view the saved *.res* file for similar information. It is usually a good idea to view the *.res* file as any run-time problems will have been listed here. The file is viewed with the default editor (notepad) as follows.

```
File ↓
     Display Res File ...
```

The default editor can be changed from the console menu:

```
Options ↓
     Change Editor ...
```

Batch Solutions

Data can be placed in batch files for later analysis. Many separate analyses can be added to a batch file for overnight or weekend runs if required. To select a file for running in batch:

```
OPERA-2d/3d ↓
     Add to batch queue ...
```

Once all the required files have been added, the batch file can be started:

```
Batch Process ↓
     Start Batch Analysis ...
```

Options to clear or list the batch queue are also available under this menu item.

Running OPERA-2d off-line

A utility program called *winbat.exe* has been included with OPERA to allow the automatic execution of analyses from simple command scripts. This removes the need to interact with the OPERA console if a large number of analyses with post processing need to be carried out. The program *winbat.exe* is run from a command/DOS prompt and requires the name of the file containing the command script as a parameter. For example:

```
winbat.exe script.batch
```

The command script contains a series of commands, that use an identical syntax to normal DOS batch commands. To see the syntax type **help *command*** in a DOS window. The commands available are listed below:

copy, del, move, rem, dir, for, mkdir, rmdir, echo, cd, set.

The **cd** and **set** commands are simple implementations only, without the full features available.

All standard executable files ending with the extension *.exe*, *.bat* and *.com* can be executed from the script. Specifying the full path to the executable is advisable. Paths or filenames that include spaces must be enclosed in quotes “ ”.

The program *winbat.exe* can be executed with a *–n* switch. This will prevent the completion message being displayed, and is useful if the winbat program is called from within a script.

Example

```
REM Set some variables
REM Set installation folder
set VFBATCH=C:\program files\Vector Fields\OPERA 8.5
REM Set local folder
set LOCALDIR=C:\opera\work1
cd %LOCALDIR%
rem set the comi file for running the pre and post-processor
copy mypre2d.comi opera2.comi
rem launch the pre and post-processor
rem note that the quotes are required to run correctly
REM /local runs in current folder
REM /min runs iconised
"%VFBATCH%\pp\pp.exe" /local /min
REM Run the analysis on the OP2 file created by the
REM Pre and post-processor command script
"%VFBATCH%\solvers\2d\st.exe" "%LOCALDIR%\mydata2d.op2"
REM prepare the post-processing command file
copy mypost2d.comi opera2.comi
REM launch the Post-processor
"%VFBATCH%\pp\pp.exe" /local /min
```

```
REM clear out the command input files
del opera2.comi
```

Please note that for users running Windows NT, 2000 and XP, it is not necessary to use *winbat.exe*. Batch scripts as above can be run directly. If running directly under NT, 2000 or XP the **cd** command should be replaced with **cd /D**

Text IO Window

The OPERA pre processor allows for data entry via the keyboard. This is achieved by selecting **MENU OFF** from the top menu bar. A text input window prompts the user to enter the appropriate commands. This window may be moved or resized just like any other window. Its contents may be scrolled, either by using the scroll bar, or with the <up-arrow>, <down-arrow>, <page-up> and <page-down> keys. If the prompt is out of sight, pressing the <return> or any other character will scroll the window back to the prompt.

The line currently being entered may be edited; the <Insert> key toggles between overstrike and insert mode, with the cursor prompt changing to indicate the mode. The <left-arrow>, <right-arrow>, <home> and <end> keys may be used to move the cursor prompt. Deletion can be done with the <backspace> key.

Previous commands that have been executed may be re-selected by pressing <shift up-arrow> or <shift down-arrow>. These commands can be edited as described above and then re-issued with the <return> key.

The number of lines in the Text IO history buffer can be controlled using:

Options ↓
Window Preferences

and set the value of **Message Screen History**.

The Text IO window can be hidden or ‘always on top’ using the top (grey) menu bar command:

View ↓
Console

Typing ^ (normally SHIFT+6) at the cursor prompt will restore control to the menus.

File Names and Extensions

Files created by the programs have extensions supplied by the programs. The extensions are upper-case for upper-case file names and lower-case for lower-case filenames. The file name extensions are as follows. Some file extensions are common between 2d and 3d programs. For example *.bh* files.

2D Files

.bh	B-H data files created by OPERA-2d/PP
.op2	data files
.mesh	mesh files
.“analysis”	results files from analysis (AC, SA, ST, TH, TR,VL or RM)
.res	log files created by OPERA-2d analysis programs
.dem	design environment data file
.emit	emitter files used for space charge problems
.comi	command files
.tracks	particle tracking files
.table	table files
.ps	postscript format graphics files
.hgl	hpgl format graphics files
.pic	internal picture format graphics files

Various log files are created during the running of the pre and post processor. These are:

*Opera2d_PP_***n**.*lp*	All user input and program responses are recorded here.
*Opera2d_PP_***n**.*log*	User input is recorded here. These can be reused as input to the program by converting to *.comi* files.

The log files are stored in a sub-directory of the project folder *opera_logs*. The file history is always retained. Each time the software is started, new *log*, *lp* etc. files are generated. Previous files are not overwritten. The history is retained by attaching a number, **n** to the files. The value of **n** is chosen to be the lowest available integer not already in use. This may lead to a situation where the lowest value of **n** is not necessarily the oldest file: for example if some older files were deleted, those values of **n** are re-used later.

COMI Files

The *.comi* file is a command file, that can be run using the $ COMINPUT command, or using **File → Commands In** menu. The file is a text file, that can be created using a standard text editor. The *.comi* files can also be used to automatically start up the interactive programs. Each interactive program always reads the appropriate *.comi* file on start up (although they are normally empty, the user can add commands if desired).

opera2.comi - The pre and post processor always reads this file first.
opera2_de.comi - The Design Environment pre and post processor always reads this file first.

BH Files

The *bh* folder in the installation folder contains sample BH data files for use with OPERA-2d and OPERA-3d. The folder also includes a *bh.index* file.

Windows Menus

In addition to the standard GUI menus, there is a set of additional menus specifically for the Windows implementation. Their function is to control the behaviour of the OPERA pre and post processor within the Windows environment.

FILE Menu

The **FILE** menu allows printing of the windows, and displaying some related files.

Selecting **FILE → Print** allows printing of all the windows, or just the graphics window. If **All Windows** is selected, then the surrounding border and the OPERA GUI menus are included in the print. If **Graphics Window** is selected, the top level menus are omitted (but any pull-down menus overlapping the graphics window are included in the print). The Windows print manager is used to perform the printing in the usual way.

The option **FILE → Display Res File** launches Windows Notepad and prompts for the file to be loaded. All files with the *.res* extension are displayed for selection.

The option **FILE → Display Emit File** is similar to above, with files having extension *.emit* displayed for selection.

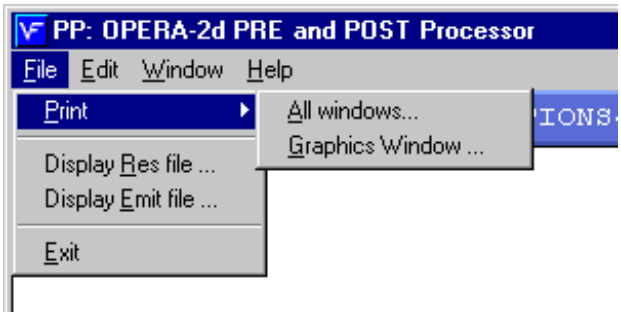


Figure 2.1 File menu including Print sub-menu

EDIT Menu

The **EDIT** menu copies the contents of the windows to the clipboard. As above, the options are for **All Windows** or just the **Graphics Window** to be copied.

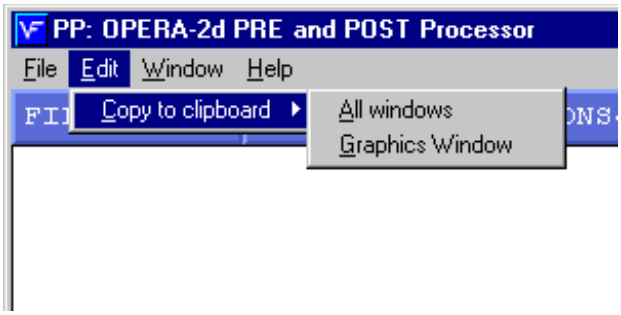


Figure 2.2 Edit menu allowing copying to clipboard

WINDOW Menu

The **WINDOW** menu controls the positioning of the Console and the Graphics windows on the screen.

The options for the Console window are:

Normal: The Console window (showing the equivalent keyboard commands on menu selection, or allowing keyboard input of the appropriate commands in place of the GUI) is on top when interacting with it, but is positioned behind the graphics window when the GUI is active.

Always on Top: The Console window is forced to always be positioned on top of all other windows.

Hidden: The Console window is deleted completely, but can be displayed again using one of the above options.

The options for the Graphics window are:

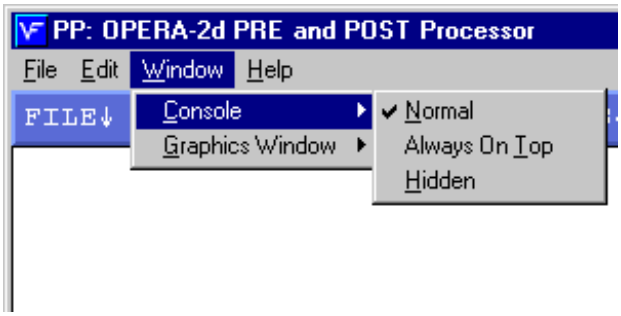


Figure 2.3 Window menu controlling positioning of Console and Graphics windows. The Console window options are shown.

`Restore`: The Graphics window is reduced in size, and can then be resized as required in the normal manner.

`Maximised`: This sets the Graphics window to the default size, where it completely fills the available space within the OPERA window.

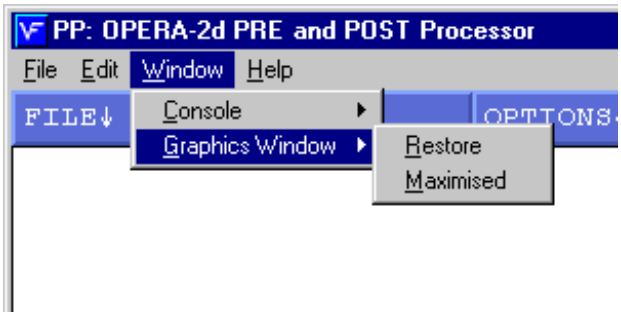


Figure 2.4 Window menu controlling positioning of Console and Graphics windows. The Graphics window options are shown.

HELP Menu

The **`HELP`** window allows access to the online help. The relevant Reference Manual is present in WinHelp format. In addition, the **`ABOUT`** option displays the current software version number, and the contact details for Vector Fields.

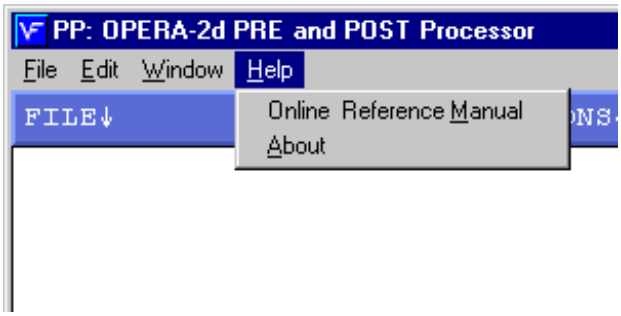


Figure 2.5 Help menu allowing online help to be displayed (WinHelp)

Chapter 3
Program Philosophy

Introduction

OPERA-2d is a suite of programs for 2-dimensional electromagnetic field analysis. In addition, there are modules for stress and thermal analysis. The programs all use the finite element method to solve the partial differential equations that describe the behaviour of fields. Such equations include:-

- Poisson's equation
- Helmholtz equation
- Diffusion equation

The solution of these equations is an essential part of design in the following areas:-

- Magnetostatics
- Electrostatics
- Time-varying magnetic fields (low frequency)

The ability to model non-linear materials is essential to these applications.

The software uses the finite element method (FEM). Since much information is required before the analysis may be performed, data entry is carried out using a powerful interactive pre processor. Using the graphical interaction within the pre processor, the model space is divided into a contiguous set of (triangular) elements. The physical model may be described in cartesian or cylindrical polar (axisymmetric) coordinates.

Once the model has been prepared, the solution is calculated using a suitable analysis module. Several modules exist for analysis of the different types of electromagnetic excitation conditions e.g. static, steady state. The analysis program iteratively determines the correct solution including non-linear effects if these are modelled.

The result may then be examined using a versatile interactive postprocessor. As with the pre processor, this is predominantly controlled by interaction through a graphical menu system. Many system variables are available for examination, including potentials, currents, fields, forces, temperature. The additional feature of user defined variables allows the solution results to be tailored to specific applications. Numerical errors due to coarse mesh definition are also analysed so that the mesh can be refined to achieve the required accuracy.

This chapter summarizes the finite element method, which is common to all the analysis programs, giving suggestions for checking and maximizing the accuracy of the results, and also gives an overview of the field equations solved and how the parameters of the pre processor can be used to access specific features of each program.

Analysis Programs

There are a number of analysis programs in the OPERA-2d Suite. They all read data prepared by the pre and post processor and create results files which can be read by the pre and post processor. The analysis programs are:

Statics:	
ST	Linear or non-linear magnetostatics or electrostatics with isotropic materials and permanent magnets.
SP	Electrostatics, including the effects of space charge from particle beams.
Eddy Currents:	
AC	Steady-state ac eddy currents with linear or non-linear materials and either current or voltage driven sources. Permeabilities can either be taken from the region data, looked up from a previous ST or TR solution, or calculated from the maximum field in the AC solution. In all cases the permeability can be complex.
TR	Transient eddy currents with multiple drives and linear or non-linear materials, coupled to external circuits.
VL	Eddy currents induced by constant velocity motion of one part of the model with respect to the rest.
RM	Rotating Motion Module: a transient eddy current module, extended to include the effects of rigid body rotation, time varying currents and coupling to external circuits.
LM	Linear Motion Module: a transient eddy current module, extended to include the effects of linear motion.
Stress Analysis:	
SA	Stress analysis, using nodal forces as input, or body force densities calculated from an earlier electromagnetic analysis.
Thermal Analysis:	
TH	Thermal analysis, using nodal temperatures as input, and element power densities calculated from an earlier electromagnetic analysis.
THTR	Transient version of the thermal analysis.

Self Adaptive mesh refinement is now available as a standard feature with the ST, AC and VL solvers.

There is also 1 utility program:

DXF	A program to translate DXF data files into OPERA-2d Command Input Files (*.comi*).

Program Philosophy

A complete problem solution with OPERA-2d consists of 3 phases: data preparation or pre processing; analysis; results display or post processing. Because the pre and post processor is one program, any modification to data can be made immediately following post processing.

The OPERA-2d Model

A large number of electromagnetic devices can be represented by a two dimensional model. This assumes that the device falls into one of the following categories:

- rotationally symmetric. There is no component of field in the azimuthal direction, and the field distribution is the same for any axial cross section. In OPERA-2d, such models are referred to as AXISYMMETRIC.
- long in one direction, with uniform cross-section over much of the length. In such devices, it is a reasonable approximation to assume that for much of the length, the field distribution over the cross section does not change and that there is no component of field parallel to the long axis. In OPERA-2d, such models have XY symmetry.

Pre Processing

The geometry of a device to be analysed by OPERA-2d is presented to the pre and post processor as a set of polygonal areas or regions on the 2d plane. One region can be a ‘background’ region which covers the whole problem space. The others are non-overlapping polygons which specify the other materials in the problem. In magnetic devices, for example, a region can represent one of the following:

- free-space;
- a conductor with a prescribed or induced current density;
- permeable material with a linear or non-linear material characteristic.

A special gap region must be defined for the interface region between the stator and rotor of rotating machines for OPERA-2d/RM. The gap region may also be used beneficially for electrical machine models with other analysis programs. See “Rotating Machines (RM)” on page 5-31.

OPERA-2d/LM models make use of a special remeshing technique that requires the user to separate the model into three groups of regions: the moving regions,

the static regions, and the regions that allow the motion. During the time stepping solution process, the moving section of the model is repositioned and a reconnected mesh between the moving and the static sections is created. The features of the LM module are fully described in Chapter “Linear Motion (LM)” on page 5-41.

Within each region, finite element mesh generation is automatic, using as input data the coordinates of the vertices and the curvatures and subdivisions of the sides. There are two classes of regions shapes: quadrilaterals and general polygons. The mesh within quadrilaterals is generated by transformation to a unit square and regular subdivision. The mesh is therefore predictable and mesh generation times are short. Elements with large aspect ratios necessary for small air gaps can be generated. The mesh within polygons is generated using an algorithm based on Delaunay triangulation. Internal nodes are added if necessary to achieve element sizes which vary smoothly across the regions and element shapes which are as near to equilateral as possible. Polygons allow large areas of space, especially near a ‘far-field’ boundary, to be meshed with a minimal number of regions.

To enable repeated structures to be defined efficiently, regions carry with them replication parameters, which create multiple copies of the region with the same material properties. Local coordinate systems are also available with each region.

Region data is entered with the DRAW command, and can be edited with the EDIT and MODIFY commands. Regions can be copied with the COPY command.

All the material properties and boundary conditions are stored with each region. The exceptions to this are the non-linear relationships between flux density and field strength which are stored as tables of values, related to the regions by material code numbers. Such BH tables are defined and edited with the BHDATA command.

Regions are also used in post processing as the smallest units of area over which integration can be performed (INTAREA command).

OPERA-2d/PP also has commands for displaying the region data, numerically and graphically (PRINT, RECONSTRUCT), generating and checking the mesh (MESH, CHECK) and READING and WRITING files. Before writing a data file, the analysis specific data should be added using the SOLVE command. No further interaction is necessary before running the analysis program, which can be started without leaving the pre and post processor.

Analysis

To provide the additional information necessary for each analysis program, the SOLVE command is run, which allows the user to define such items as convergence tolerances, output time points, non-linear iteration type, etc. A data file containing the model and the analysis specific data is created and the analysis is started. The SP solver also requires the creation of a separate data file that defines the characteristics of the emission surfaces before the analysis can be run. Once the analysis has started no further user interaction is required. The programs create results files which contain a copy of the data and the solution(s), as well as a log file containing diagnostics.

Post Processing

The pre and post processor, OPERA-2d/PP, can READ files of results from the analysis programs, display and process the solutions. The solutions consist of the nodal values of potential, and element values of current or charge density and permeability or permittivity. Any simple field quantity (potential, field intensity, flux density, current density, etc.) can be displayed at POINTS, along LINES or as CONTOUR plots over regions. Algebraic expressions of such field quantities can also be used.

Further processing can take the form of integrations along lines or over regions giving values for forces, stored energies etc., or particle trajectory calculations.

The first request for a field value from the post processor prompts the program to perform ‘field averaging’. The finite element method used gives potential derivatives which are discontinuous from one element to the next. The field averaging process finds the average from each element which surrounds a node and applies that average value to the node. (Physical discontinuities are maintained.) The averaged values are used in all the post processing commands, but some can also access the un-averaged values. Comparison of the averaged and un-averaged fields allows an estimate of the local and global errors in the solution. These error estimates are calculated following the field averaging and can be displayed in the same ways as the fields.

The electromagnetic fields, **B** and **H**, can also be calculated by integration of the magnetisation and current densities in each permeable or current carrying element. This feature makes it possible to calculate very accurate results for some demanding applications, as well as in the far field.

Stress and Thermal Analysis

It is often important to know the mechanical effects of electromagnetic forces and the heating effect of currents. For this purpose, OPERA-2d has a stress analysis program and two thermal analysis programs, which can take as input body force densities or power densities calculated from an electromagnetic analysis and return displacements and stresses or temperatures.

Chapter 4
Getting Started

Starting the OPERA-2d Pre and Post Processor

OPERA-2d and associated operating system software should be installed as described in the installation notes. Once this has been completed, the program may be started.

Windows

In Windows launch OPERA-2d pre and post processor from the OPERA Console, which can be activated by pressing on the VF Icon.

Unix

To launch OPERA-2d in UNIX, type **`opera`**[1] from the operating system prompt and the initial text options for OPERA-2d modules are shown[2]. The pre and post processor is selected by typing **`PP`**. OPERA-2d then carries out a number of initialisation steps. The type of graphics display must be given as the **`SCREEN`**.[3]

```
Graphics options: SCREen, FILE, BOTH or NONE?>
```

Type

```
SCREEN
```

to select the graphics screen.

---

1. If this does not work, ask your system administrator to set up an alias for **`opera`**.
2. If your system has both 2D and 3D Vector Fields software installed, you will be asked to select 2D.
3. the environment variable VFGRAPHICS can be used to pre-set the response to **`SCREEN`**.

Pre Processing

The Pre and Post Processor Environment

Upon entering the pre and post processor environment, a graphics window with a command menu (at the top of the window) is created, as illustrated in Figure 4.1. The graphics screen is used to display model geometry. The initial text window is used to display OPERA-2d messages, prompts and command line input.

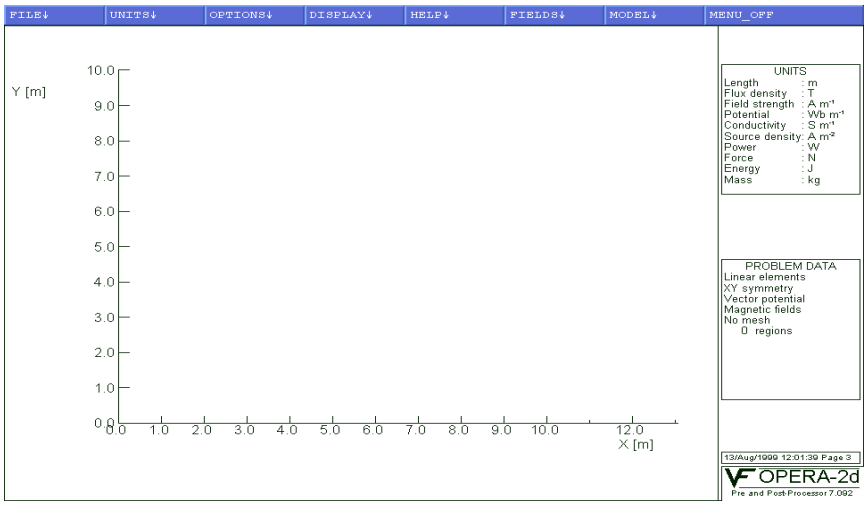


Figure 4.1 Command menu, graphics screen and display text

Changing the Units

The default set of units are S.I. and the axes are displayed in metres. The default units may not always be convenient, as is the case here. In this example, a more suitable length unit is centimetres. To make this change click-on **UNITS** in the main menu by:-

- Placing the cursor over the **UNITS** menu option. This will highlight the option.
- Pressing the left mouse button.

The sequence of menus is shown in Figure 4.2, i.e. select the **UNITS** item from the top level menu, select the **Length unit** item from the next level menu and then set the units by clicking on the **Centimetre** item. This will cause the button to be depressed as seen in the figure.

Figure 4.2 Selecting the length units

This sequence of selections will be represented in these notes by the following format

UNITS ↓
Length unit → **Centimetre**

Select

UNITS ↓
Length unit → **Return**
UNITS ↓
Return

to close the sub-menu and return to the top level command menu. To Refresh the graphics screen select

DISPLAY ↓
Refresh

The axes are now scaled in centimetres.

If unwanted sub-menus are selected in error, hitting the Return option closes the sub-menu. Alternatively, placing the mouse outside the sub-menus and hitting the left mouse button will close the menus (except in the case of pick menus[1]), as will pressing the **ESC** (escape) key.

Building a Model

Figure 4.3 shows a three dimensional cut view of a coil, wound around an iron “E” shaped former positioned close to a circular metallic disc.

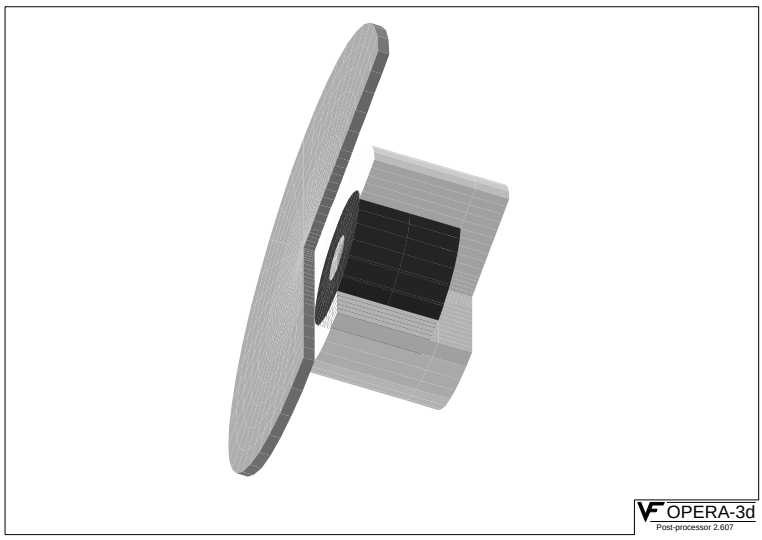


Figure 4.3 Three Dimensional View of Coil Former and Disc

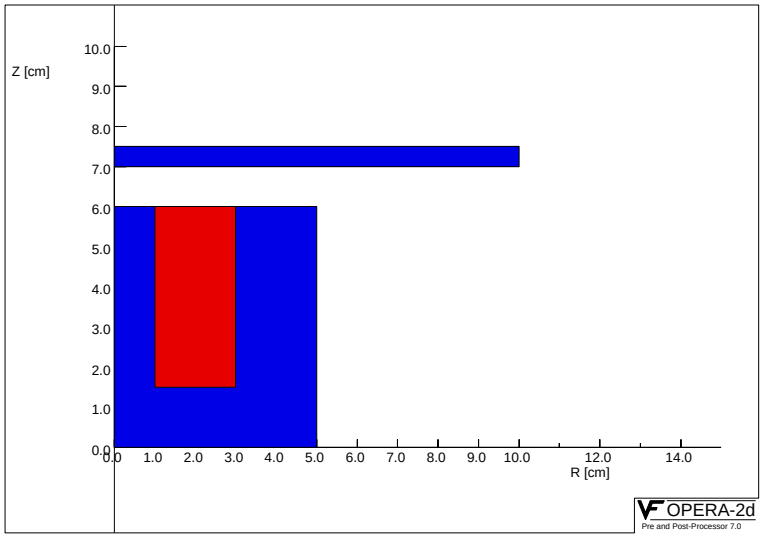


Figure 4.4 Two Dimensional View of Coil Former and Disc

1. These are menus where the user is required to pick items from the graphics display. This is recognizable by the cross hair cursor that is displayed.

Using the axial symmetry of the coil, former and disc, it is possible to model this using a two dimensional section. The two dimensional representation is shown in Figure 4.4.

Each component of the model (including air) must be defined as a region (or regions). A region is the basic OPERA-2d building block for model construction. It consists of:-

- A polygon which defines the geometric properties of the component. Once the model has been completed, this is subdivided into finite elements by the automatic mesh generator, according to the number of subdivisions assigned on each side of the polygon.
- a set of characteristics which define the material properties of the component. These properties include magnetic and electric properties of the region. These characteristics are specified at the time the region is drawn but may be modified later.

Setting the solution symmetry and potential

The problem is axi-symmetric with the axis of symmetry displayed as the vertical direction on the screen. To set this, select the series of sub-menus

MODEL ↓
Solution Type → **Axi-symm and potentials**

Upon selection of the Axi-symmetry option, different types of solution potentials appear. Select

MODEL ↓
Solution Type → **Axi-symm and potentials** → **Modified r * A**

This will cause the units to be changed and an appropriate message is conveyed to the user. Click on **Continue** or press any key to clear the message box. The menus can be seen in Figure 4.5.

This has now set the model to use axi-symmetry i.e. an R-Z coordinate (cylindrical polar coordinate) system. The solution potential of MODIFIED R*A is a modified vector potential that should be used to improve accuracy near the axis in axi-symmetric problems.

Close this menu by selecting **Return** three times.

Refresh the display as before and the axes change to display the R-Z coordinate system.

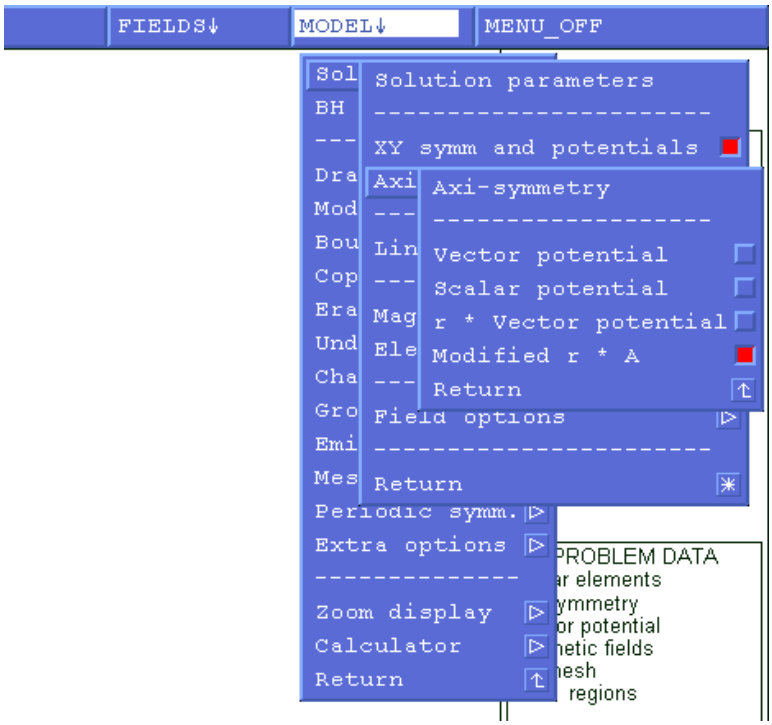


Figure 4.5 Setting the solution potential

Drawing regions - modelling the disc

In this example the first step is to create the model of the disc. The material properties will be defined by selecting

MODEL ↓
Draw regions → **Region defaults ...**
... material type

With this menu selection, the parameter box shown in Figure 4.6 is displayed.

Within this parameter box, it is possible to edit the data to set the required material properties for the disc.

- The <back-space> key can be used to delete values within the box.
- The <return> key will move the focus down to the next item in the box. If the focus is on the **Accept** button, the values within the box will be accepted, and the box will be closed.

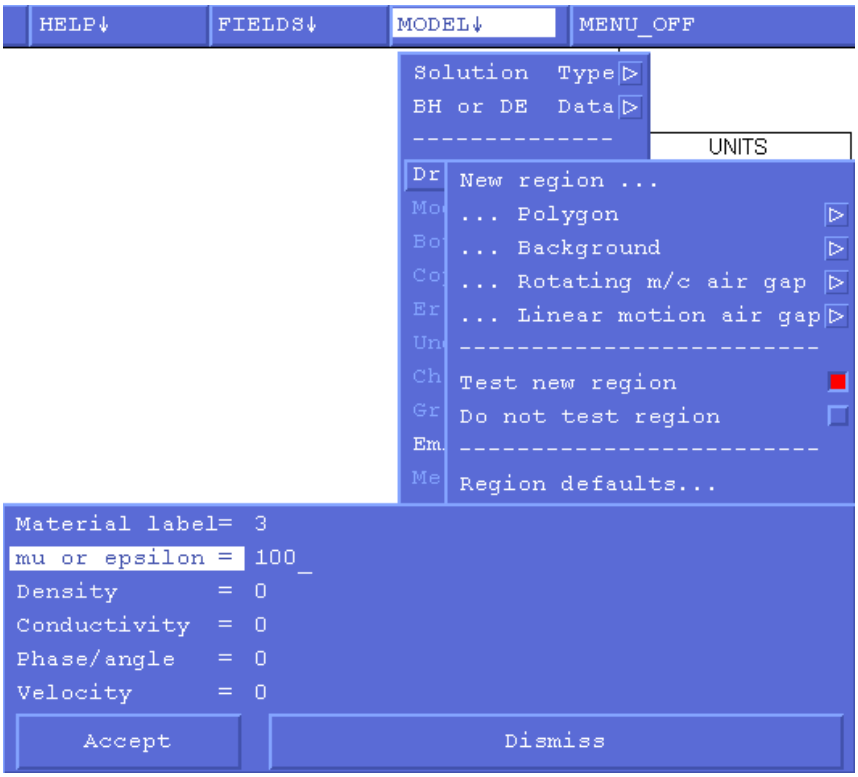


Figure 4.6 Setting the default material properties

- If a mistake is made, the **ESC** (escape) key will remove the box, and no command will be issued. This can also be accomplished by selecting **Dismiss** using the mouse.
- The <up-arrow> and <down-arrow> keys may be used to change the focus between different items in the parameter box. The mouse can also be used to select the item to be edited by positioning over the item and pressing the left button.

For this model, the disc is to have a material label of 3, implying that it is a ferromagnetic material. Also, it is to have a relative permeability of 100. Enter this information in the parameter box, i.e. setting the material label to **3** and mu to **100**. The correct settings are shown in Figure 4.6.

Once complete, move down and select the **Accept** button.

The first region geometry may now be entered as a polygon defined by a set of points joined by lines. To position the first point in the polygon (at 0,7), select

Draw regions → New region...
... Polygon → XY input

as shown in Figure 4.7.

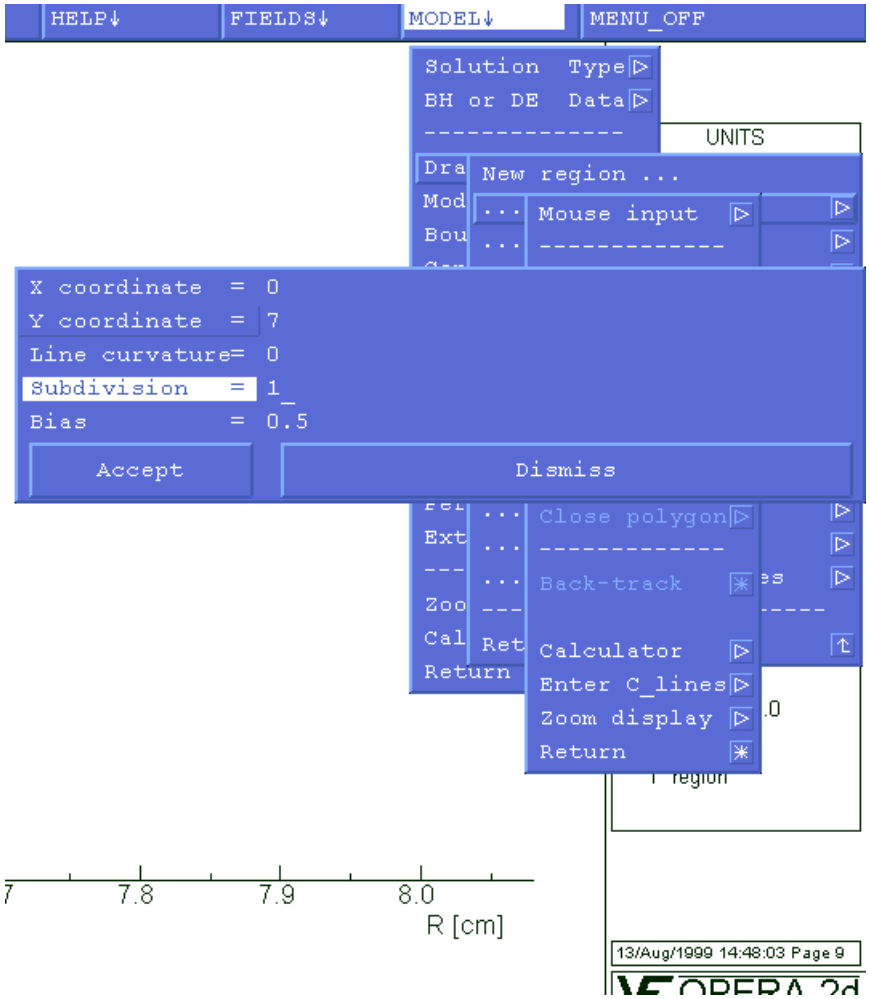


Figure 4.7 Polygon data entry menu

The parameter box is now displayed ready for numeric information. The defaults for the X coordinate (0), bias, curvature and subdivisions are left unchanged. Only the Y coordinate needs to be altered, and this should be set to 7.

Select the **Accept** button, when the data has been entered into the box. A point is drawn at coordinate (0,7).

The information that has been entered is:

- The X and Y coordinates of the point being defined.

- The Curvature of the side (equal to 1/radius). This allows curved edges to be modelled. The default value of 0 is used for straight lines. For the first point, no side is created so this data is not used.
- The number of subdivisions for the side being created. This is used when generating the finite element mesh. Again, for the first point, this data is not used as no side is being created.
- Bias of the side subdivision. This allows the subdivisions to be concentrated or graded to either the start or end of the side being created. A value between 0 (the subdivisions at the start of the side are smaller) and 1 (the subdivisions are concentrated at the end of the side) should be entered. A value of 0.5 is the default, giving a uniform distribution of subdivisions along the side.

The second point (and the line or side joining it to the previous point) may be positioned in the same way, by selecting

Draw regions → **New region...**

... Polygon → **XY input**[1]

Again the parameter box is displayed and the following should be entered:-

X coordinate	=	**10**
Y coordinate	=	**7**
Line curvature	=	**0**
Subdivision	=	**20**
Bias	=	**0.5**

Accept Dismiss

and **Accept** selected. This time the values of curvature, subdivision and bias are significant as a side is being created.

A second point and the first side of the disc are now displayed, as shown in Figure 4.8.

An alternative method of creating a line which is either horizontal or vertical is achieved using the move feature. In this case a vertical move is to be chosen by

1. Note: It is important when defining the polygon geometry, that the **New region ... Polygon** menu is not closed until the polygon is complete. If this happens, it will be assumed that you wish to close the polygon, before the definition is complete. With fewer than 3 points defined, the data will be lost, otherwise a message box will appear asking if you wish to keep the unfinished polygon. Select **No** and start the polygon definition again.

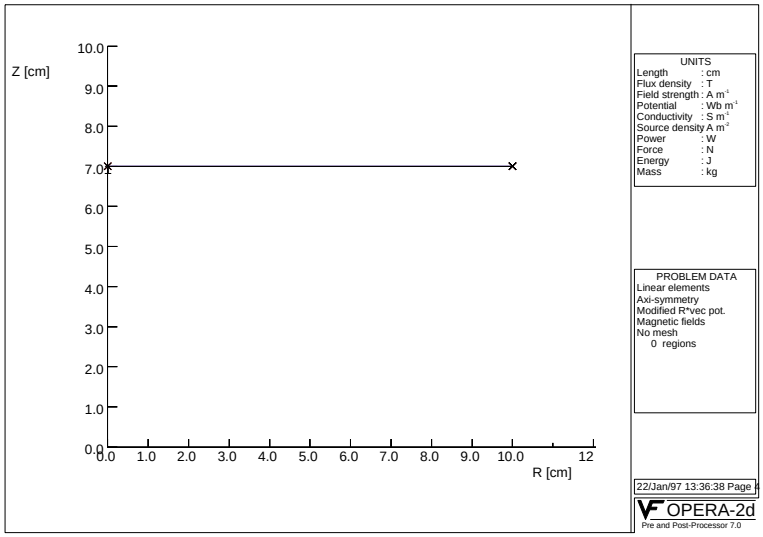


Figure 4.8 The first side of the disc

Draw regions → New region ...
... Polygon → Y move

The parameter box in Figure 4.9 is now displayed and should be completed as shown before selecting the **Accept** button.

Y displacement	=	0.5
Line curvature	=	0
Subdivision	=	3
Bias	=	0.5

Accept Dismiss

For the last point, select

Draw regions → New region ...
... Polygon → X move

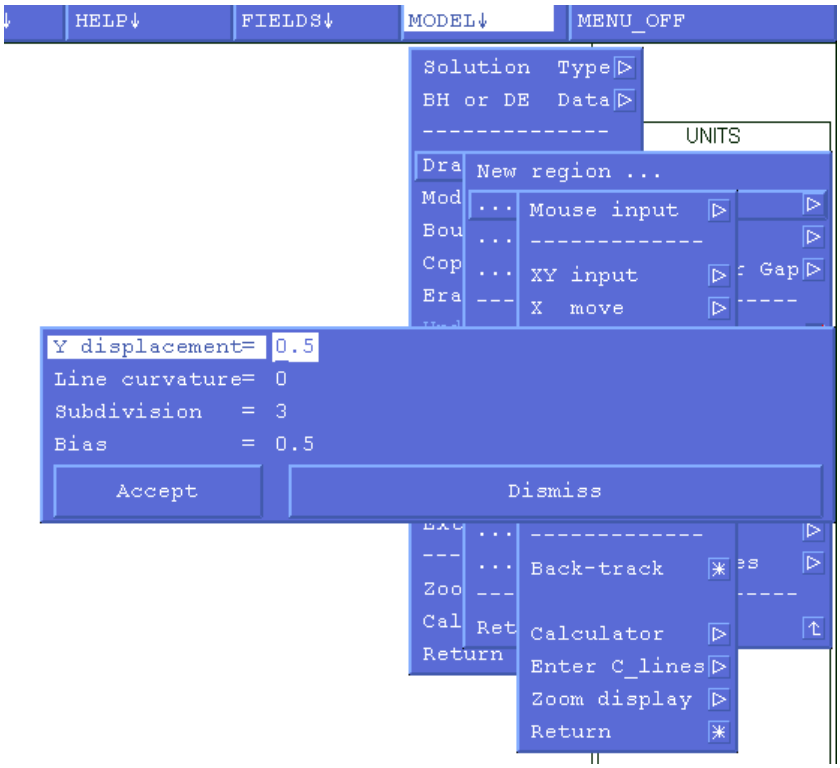


Figure 4.9 The completed Y move parameter box

A parameter box is displayed and the following should be entered

X displacement	=	-10
Line curvature	=	0
Subdivision	=	20
Bias	=	0.5
Accept		Dismiss

and **Accept** selected.

To complete the disc region, the polygon must be closed.

Draw Region → New region ...
... Polygon → Close polygon

Complete the parameter box as shown below

Subdivision	=	3
Line curvature	=	0
Bias	=	0.5

Accept Dismiss

click-on **Accept**

This draws in the final side of the region and the region is filled. The cross-section of the disc is shown in Figure 4.10.

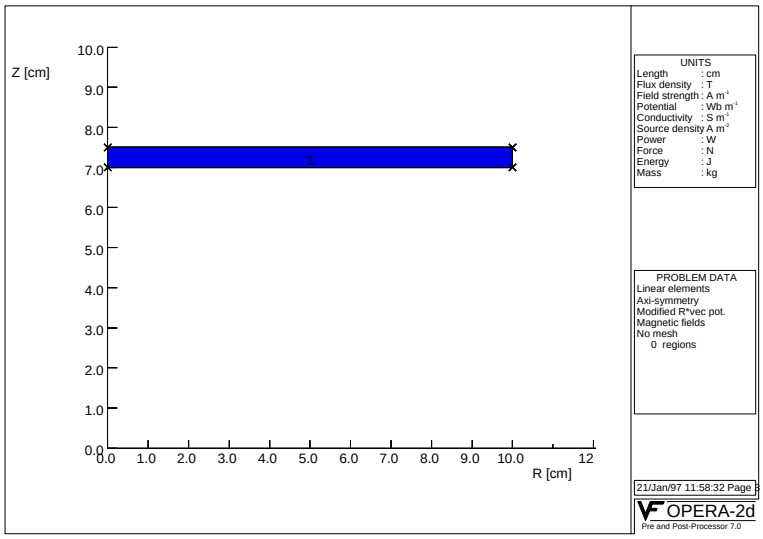


Figure 4.10 The display of the completed disc region

Each region is identified by a number displayed at the centre. The region shown in Figure 4.10 is identified by the number 1. Now close the submenus by selecting **Return** twice.

Drawing regions - modelling the coil

The next region to be specified is the conductor cross-section. The conductor will be modelled as a single region, even though the actual coil has multiple turns[1]. The fields produced by using such a model are a good representation of the actual fields. The same techniques used when building the disc will be applied here.

1. Setting the conductivity of the material to zero models a multi-turn coil

Initially the material properties of the region will be set. The settings are a material label of 1, a permeability of 1 and a current density of 2e6Am^{-2}. Then coordinates of the region and the side data will be entered.

To set the material properties select

MODEL ↓
Draw regions → **Region defaults ...**
... material type

and complete the parameter box:

Material label	=	**1**
Mu or epsilon	=	**1**
Density	=	**2e6**
Conductivity	=	**0**
Phase/angle	=	**0**
Velocity	=	**0**
Accept		Dismiss

and then **Accept**

To enter the coordinates of the region, select the following menu items and then complete and **Accept** the parameter boxes as shown below.

Draw regions → **New region ...**
... Polygon → **XY input**

X coordinate	=	**1**
Y coordinate	=	**1.5**
Line curvature	=	**0**
Subdivision	=	**3**
Bias	=	**0.5**
Accept		Dismiss

Draw regions → **New region ...**
... Polygon → **Y move**

Y displacement	=	**4.5**
Line curvature	=	**0**
Subdivision	=	**10**
Bias	=	**0.5**
Accept		Dismiss

Draw regions → New region ...
...Polygon → X move

X displacement	=	2
Line curvature	=	0
Subdivision	=	6
Bias	=	0.5
Accept		Dismiss

Draw regions → New region ...
... Polygon → Y move

Y displacement	=	-4.5
Line curvature	=	0
Subdivision	=	10
Bias	=	0.5
Accept		Dismiss

Draw regions → New region ...
... Polygon → Close polygon

Subdivision	=	6
Line curvature	=	0
Bias	=	0.5
Accept		Dismiss

The polygon is now drawn on the screen, giving the display shown in Figure 4.11.

Now **Return** from the draw polygon menu.

Drawing regions - modelling the coil former using existing points

The region representing the “E” shaped former is to be drawn next.

The material properties will be set first, giving this region a material label 4 with a permeability of 300. To do this select

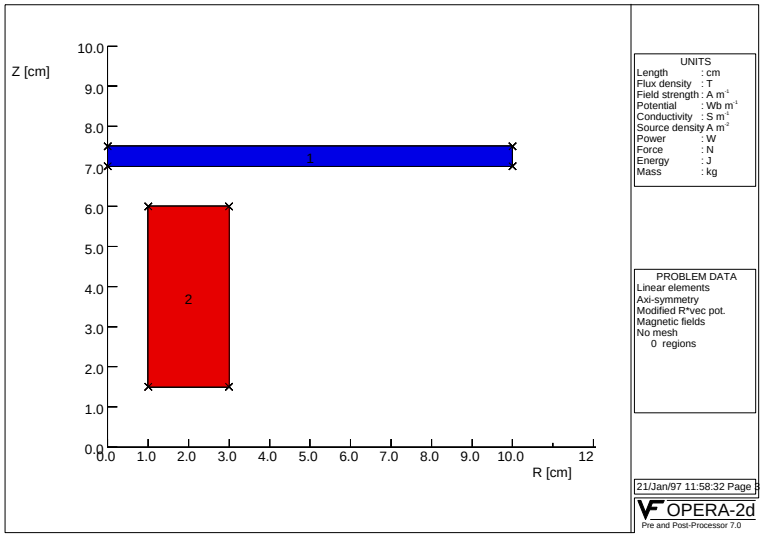


Figure 4.11 The display of the first two regions

MODEL ↓

Draw regions → **Region defaults ...**

... material type

Material label	=	**4**
Mu or epsilon	=	**300**
Density	=	**0**
Conductivity	=	**0**
Phase/angle	=	**0**
Velocity	=	**0**

Accept Dismiss

and **Accept**.

For the point definition, a new point will be defined first, using XY input as before. Selection of existing points from the conductor vertices will be used to efficiently enter the next 4 points. Three more new points will be defined before closing the polygon.

To do this select

MODEL ↓

Draw regions → **New region...**

... Polygon → **XY input**

X coordinate	=	**0**
Y coordinate	=	**6**
Line curvature	=	**0**
Subdivision	=	**6**
Bias	=	**0.5**

Accept Dismiss

and **Accept** the parameters. This places the first point of the “E” shaped former at coordinates (0,6).

The mouse will now be used to select the next 4 points. Now select

Draw regions → **New region ...**

... Polygon → **Mouse input** → **At old point**

as shown in Figure 4.12.

Moving the cursor near to the existing point at (1,6)[1], hit the left mouse button. This selects the nearest point at (1,6) as the second point of the “E” shaped former. The side joining the first and second points is also drawn.

Now continue using the cursor and mouse left button to select the next 3 points from the conductor geometry by clicking near the following coordinates:

1, 1.5
3, 1.5
3, 6

The side data will recalled from the existing conductor sides when adding these points. This ensures continuity of the mesh. Now select **Return** to return to the draw polygon menu before defining the last 3 points using X and Y moves.

1. It is not necessary to be exact when positioning the mouse. The nearest point to the cursor will be selected. If a mistake is made the back-track option may be used to undo a point.

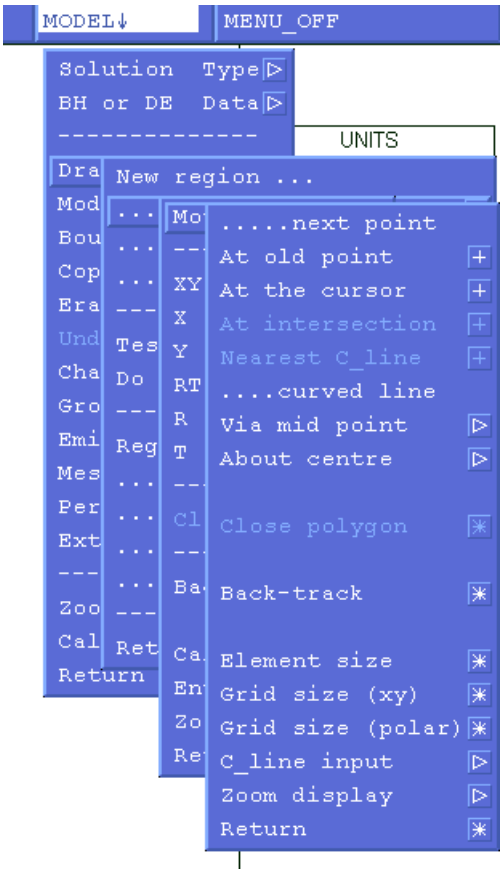


Figure 4.12 Selecting points by mouse

To draw the remaining sides of the “E” shaped former, select

Draw regions → New region ...

... Polygon → X move

X displacement	=	**2**
Line curvature	=	**0**
Subdivision	=	**10**
Bias	=	**0.5**

Accept Dismiss

and **Accept**. Then select[1]

1. After filling in the required information, select the **Accept** button in each case.

Draw regions → New region ...
... Polygon → Y move

Y displacement	=	**-6**
Line curvature	=	**0**
Subdivision	=	**16**
Bias	=	**0.5**
Accept		Dismiss

Draw regions → New region ...
... Polygon → X move

X displacement	=	**-5**
Line curvature	=	**0**
Subdivision	=	**16**
Bias	=	**0.5**
Accept		Dismiss

Draw regions → New region ...
... Polygon → Close polygon

Subdivision	=	**16**
Line curvature	=	**0**
Bias	=	**0.5**
Accept		Dismiss

This gives the display shown in Figure 4.13.

Drawing regions - modelling the surrounding air

For most EM analysis problems, it is necessary to model the air regions surrounding the problem. The air regions between the magnetic regions may be drawn in a similar way using polygons that fit between and around the 3 regions drawn so far.

However an alternative and more convenient method is also available. This method defines a *background region* [1]. This is a region which may be simply defined and overlaid on the existing model. The spaces between existing regions are automatically meshed. This method will be used here.

1. Only one background region is permitted in the model

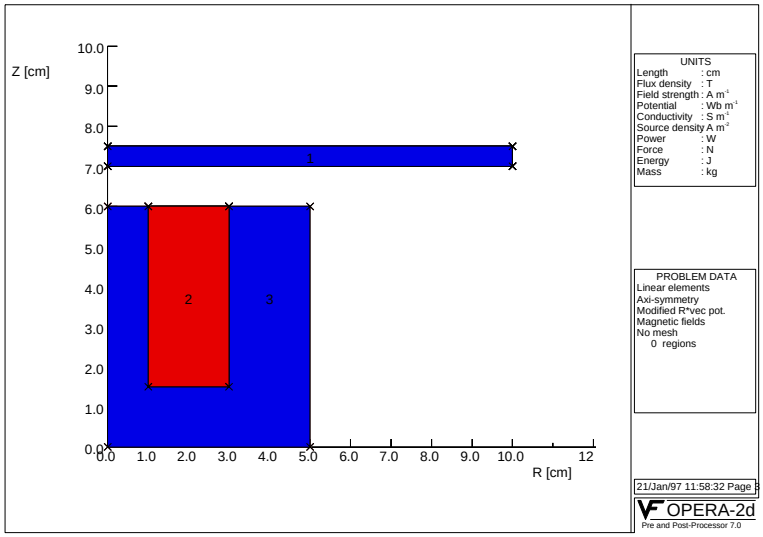


Figure 4.13 The completed "E" shaped former

Initially set the material properties to air by

MODEL ↓

Draw regions → **New region ...**
... Polygon → **Return**

Draw regions → **Region defaults ...**
... material type

Material label	=	**0**
Mu or epsilon	=	**1**
Density	=	**0**
Conductivity	=	**0**
Phase/angle	=	**0**
Velocity	=	**0**

Accept Dismiss

The background region is entered in the same way as other polygon regions except that the **Background** option must be selected as shown in Figure 4.14. In this case the XY input followed by X and Y moves will be used.

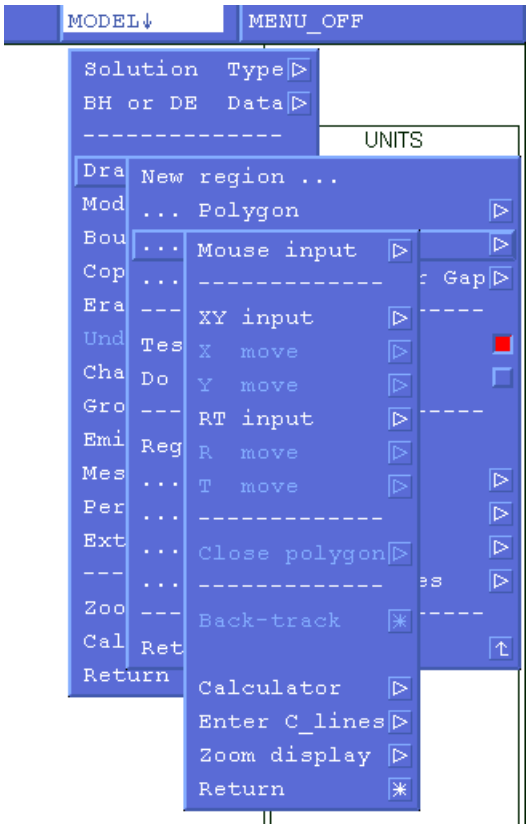


Figure 4.14 Drawing the background region

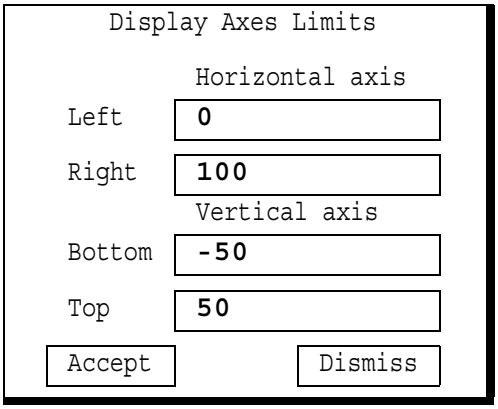
Draw regions → New region ...
... Background → XY input

X coordinate	=	0
Y coordinate	=	50
Line curvature	=	0
Subdivision	=	16
Bias	=	0.5

Accept Dismiss

The display is not large enough to see the new point. To increase the display area select

Draw regions → New region ...
... Background → Zoom Display → Numerical axes limits



as shown in Figure 4.15

For the X move, biasing will be used to keep the mesh more concentrated near the Z axis (i.e. a number less than 0.5).

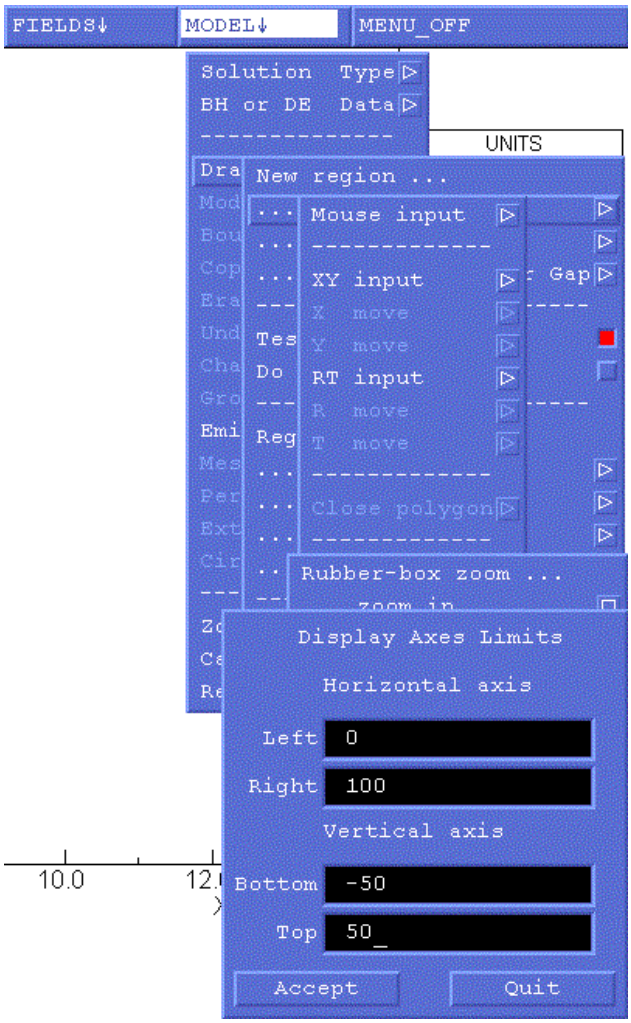


Figure 4.15 Re-sizing the display

Draw regions → New region ...
... Background → X move

X displacement	=	50
Line curvature	=	0
Subdivision	=	10
Bias	=	0.1

Accept Dismiss

Draw regions → New region ...
... Background → Y move

Y displacement	=	**-100**
Line curvature	=	**0**
Subdivision	=	**15**
Bias	=	**0.5**
Accept		Dismiss

For this “X-move” back towards the Z axis, the biasing will be greater than 0.5.

Draw regions → New region ...
... Background → X move

X displacement	=	**-50**
Line curvature	=	**0**
Subdivision	=	**10**
Bias	=	**0.9**
Accept		Dismiss

Draw regions → New region ...
... Background → Close polygon

Subdivision	=	**25**
Line curvature	=	**0**
Bias	=	**0.5**
Accept		Dismiss

As the polygon is closed, a test is performed to see whether it intersects any other regions. In this case it intersects both the disc and the coil former. A message box appears indicating that points are being added to allow continuity across the different region boundaries. Click on **Yes.**

Only one background region may be used in a model. Clear this next message by pressing any key.

Return to the **Model** menu by selecting **Return** twice.

The geometry of the model is now complete, but the mesh should be checked to ensure that it is suitable and the boundary conditions for the problem must be set.

Generating and viewing the mesh

The finite element mesh will be generated automatically from the subdivision and bias data of the regions by

MODEL ↓
Mesh generator → **Generate mesh**

The mesh generator now operates on the model data and reports on the size of the mesh in a message box. The message box also reports on the checks carried out by OPERA-2d on the model. Two warnings are given indicating that boundary conditions have not yet been applied and that no BH curve has been assigned to Material 4. Clear the message box to leave the model outline displayed[1]. Close the menu using **Return**

The mesh can be seen by setting the display option to include the FE (finite element) mesh as shown in Figure 4.16, i.e.

Figure 4.16 Displaying the finite element mesh

1. The outline should be checked carefully. Region boundaries between materials of the same type are not shown, boundaries between different materials are shown in white and external boundaries are red. If there are any red lines within the interior of the model, the mesh is not continuous and **must** be corrected before an acceptable solution can be found. If this needs to be carried out a message is displayed stating that a number of closed loops have been found.

DISPLAY ↓
-Mesh (*toggles to +Mesh)*

followed by **Refresh** giving the display shown in Figure 4.17.

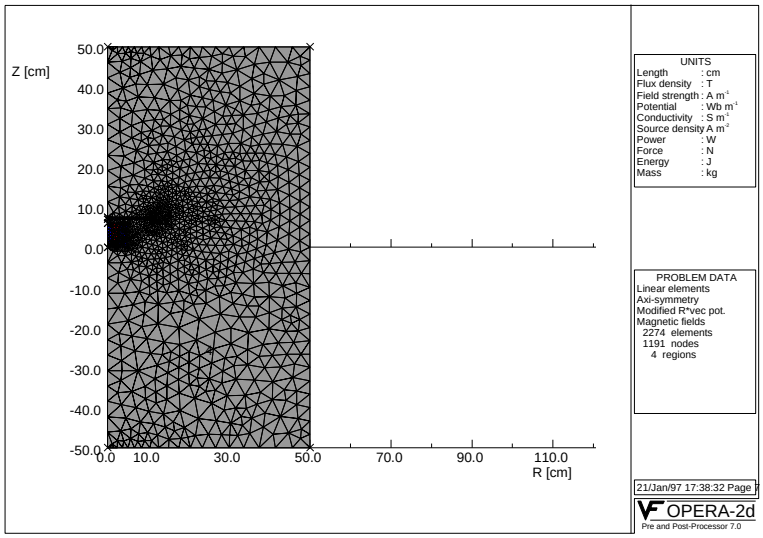
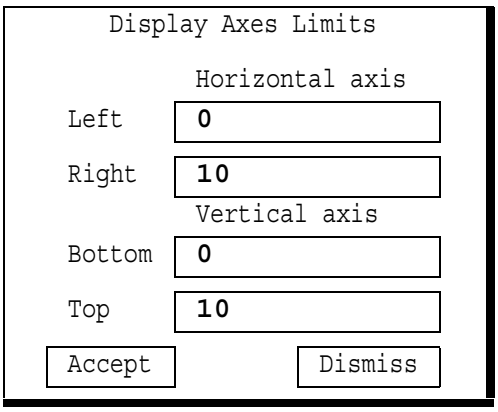


Figure 4.17 The mesh within the background region

To check the mesh around the coil former and disc it is necessary to resize the display to be closer to the area of interest. Select

DISPLAY ↓
Axes limits



You can also overdraw the axes on the mesh by selecting

DISPLAY ↓

Axes → **Overdraw axes**

and subsequently refreshing the screen. It can be seen that the mesh is not very good in the air gap, close to the Z axis. This will be improved by

MODEL ↓

Modify regions → **Modify sides ...**

... by picking → **All properties...**

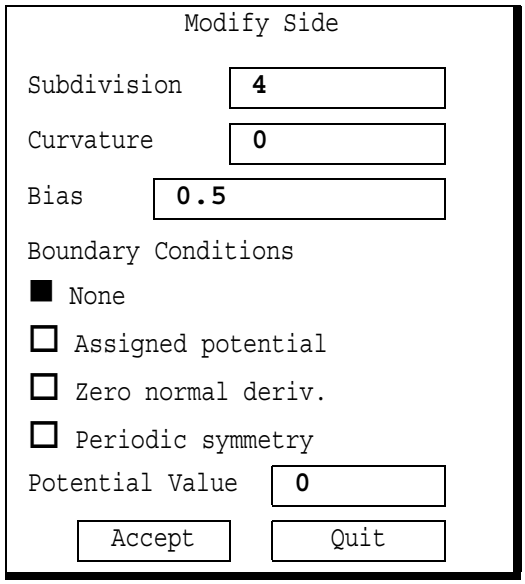
...of one side

Select the side of the background region by positioning the cursor *inside* the background region near

0, 6.5

and clicking the left mouse button.

Change the number of subdivisions in the box to 4 as shown below.



and press **Accept**. The screen will be redrawn without the mesh as this modification has destroyed the mesh data.

Now modify the top side of the coil former by positioning the cursor *inside*[1] the coil former. Select

1. The side selected will be the nearest side from the region in which the cursor is placed. The sides of the coil former will not be found if the cursor is in the background region.

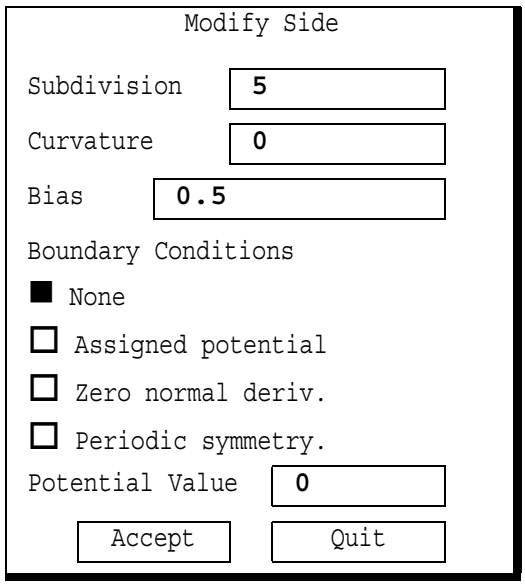
MODEL ↓
Modify regions → **Modify sides ...**
... by picking → **All properties...**
...of one side

near

0.9, 5.9

and click the left mouse button.

Change the number of subdivisions in the box to 5 as shown below.



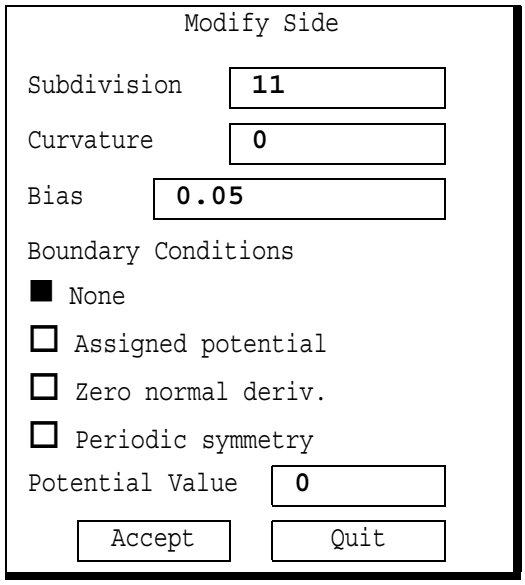
Finally modify the sides of the background region above the disc by selecting

MODEL ↓
Modify regions → **Modify sides ...**
... by picking → **All properties...**
...of one side

and clicking in the background region near

0, 10

to change the bias to

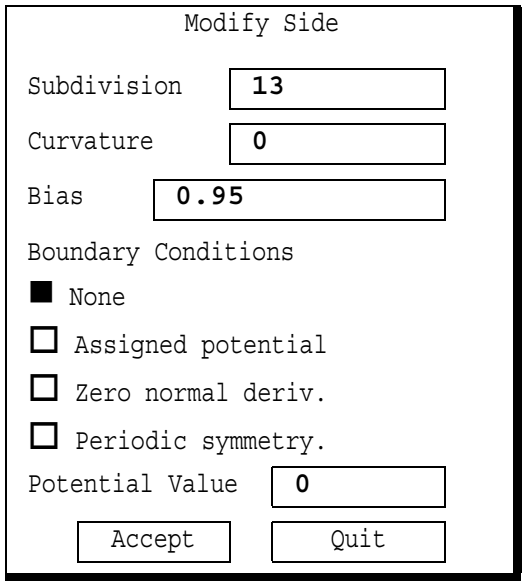


and below the coil former by selecting

MODEL ↓
Modify regions → **Modify sides ...**
... by picking → **All properties...**
...of one side

and clicking near

0, -2



For these sides, only the bias has been changed to grade the subdivisions and make them smaller in the area of interest i.e. near the coil former.

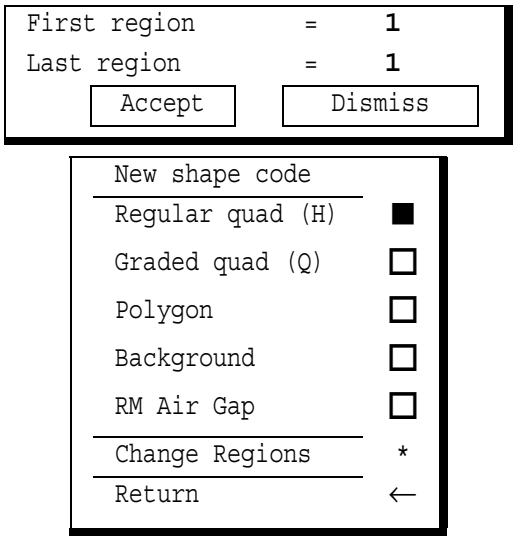
MODEL ↓
Modify regions → **Modify sides ...**
... by picking → **Return**

Modify regions → **Return**

The mesh within the disc is quite poor. This is because the mesh generator tries to form equilateral elements, which means that only one element will be generated through the thickness of the disc. To improve this, quadrilateral regions can be changed to give a regular grid of elements. The disc will be changed to a "H" region. This type of quadrilateral requires the subdivisions on opposite faces to be the same so that a regular grid of elements can be constructed.

To change to shape "H", select

MODEL ↓
Change regions → **Region Numbers**



and select **Regular Quad (H)** followed by the **Change regions** to make the change, as shown in Figure 4.18

The mesh can be generated again by selecting

MODEL ↓
Mesh generator → **Generate mesh**

and refreshing the display. The complete mesh is shown in Figure 4.19.

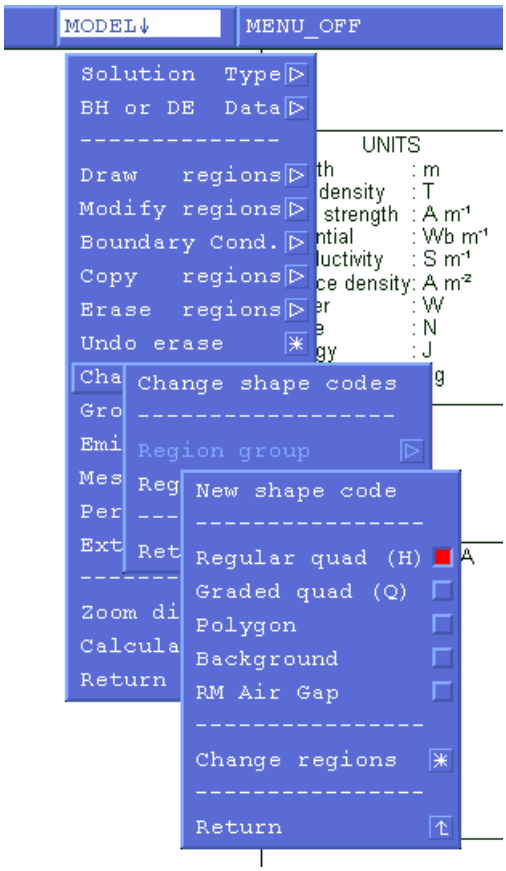


Figure 4.18 Changing the region shape codes

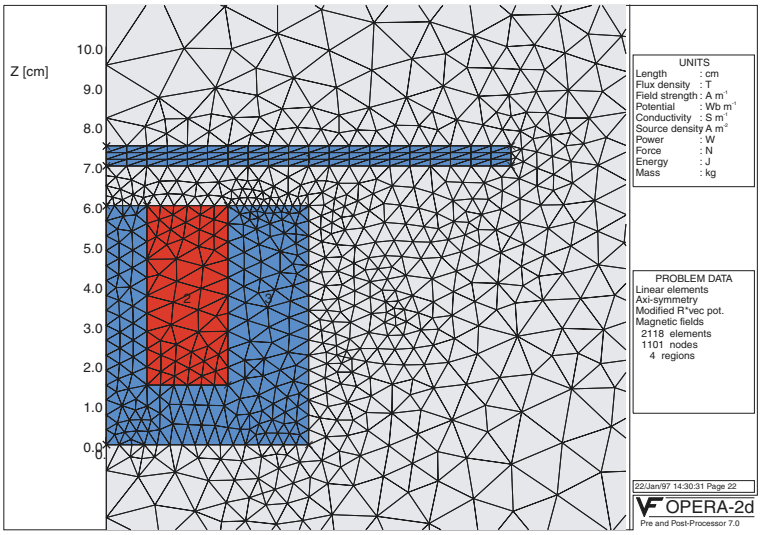


Figure 4.19 The mesh within the background region

Setting boundary conditions

Switch the mesh display option off by selecting

DISPLAY ↓
+Mesh *(toggles to -Mesh)*

followed by **Refresh**.

The boundary conditions that are to be set are such that the flux will not cross the Z axis. This is achieved by setting the normal component of **B**, on this boundary, to zero. The far field boundaries will also be set with this condition, although the effect of this boundary condition should be negligible if the boundary is at a large distance from the device.

These boundary conditions must be set by selecting

MODEL ↓
Boundary Cond. → **Vector pot** → **B normal = 0**

Hit the left mouse button near the following coordinates, inside the model regions, to select each of the different sides that exist along the Z axis:-

0.1, 10
0.1, 7.25
0.1, 6.5
0.1, 3
0.1, -1

To select the far-field boundaries, resize the screen to the previous larger limits by

Boundary Cond. → **Vector pot** → **Zoom display** → **Bounding box**

The screen will resize to the larger limits. Choose the boundary condition to be applied and then select the remaining 3 sides by:

Boundary Cond. → **Vector pot** → **B normal = 0**

25, 49
49, 0 (press F1 to temporarily hide the menus)
25, -49

Restore the menus by pressing F1 again and close the boundary conditions sub-menu with **Return** twice.

Specification of BH Curve Data

Although the present problem deals with materials of linear permeability (100 for the disk and 300 for the former, as specified earlier) the software still requires a BH curve to be assigned to each material, other than air and coils. This BH curve is used to inform the analysis module that the coercive force is zero in this instance. The software automatically assigns a default BH curve to material 3, but no curve to material 4 (hence the warning when a mesh was first generated). As the BH curve will have no effect on the linear solution, the default curve can also be assigned to material 4. This can be done as follows:

Select

MODEL ↓
BH or DE Data. → **BH / DE Editing** → **Material 3**

A message appears informing the user that the BH curve for Material 3 will be displayed and that Material 3 has isotropic properties. Clear the message and the BH curve data will appear. Save the curve as *default.bh* by selecting

BH or DE Data → **BH / DE Editing** → **Material 3** → **Store in File**

and entering the filename in the designated box. Subsequently hit **Return** and select **Material 4.** Clear the information message and select the option

BH or DE Data → **BH / DE Editing** → **Material 4** → **Load from File**

Load the filename *default.bh* and hit **Return** twice to accept all selections.

Having performed all necessary refinements to the model, the mesh can be generated by selecting,

Mesh generator → **Generate mesh**

As before, the mesh generator operates on the model data and reports on progress in a message box. Clear the box by hitting any key or the mouse button. If no errors or warnings are given, the model will satisfy the analysis module requirements. Any errors must be corrected before starting the analysis.

OPERA-2d displays an outline of the model after completing the mesh generation, as shown in Figure 4.20.

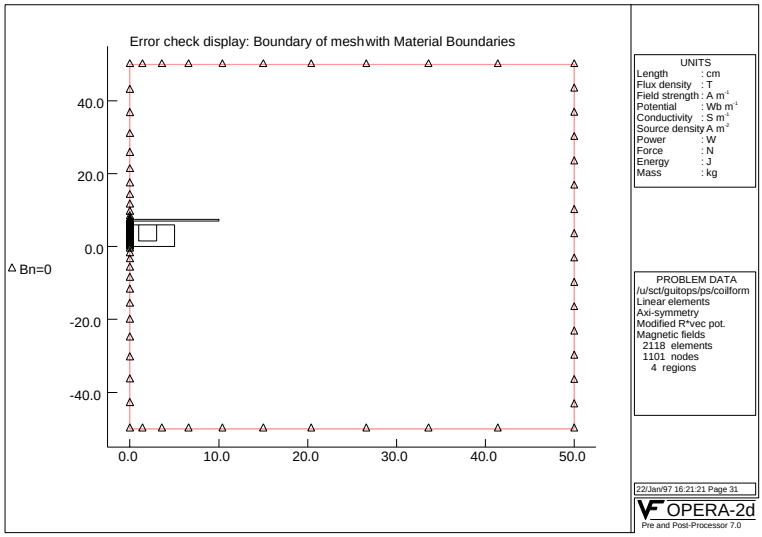


Figure 4.20 The completed mesh generation display

Solving the Problem

Solution Methods

A solution method is needed for the example problem. The analysis programs available[1] are as follows:-

- The static analysis module (ST)

  This handles both magnetostatic and electrostatic problems. It assumes that the excitation current is constant with time, or allows excitation by permanent magnet. The material properties can be non-linear (i.e. the permeability may be a function of the field strength at each point in the material). This also allows adaptive refinement of the mesh to reduce the error in the solution by improving the mesh discretisation.

- The steady-state ac analysis module (AC)

  This assumes that the excitation current or voltage has a sinusoidal waveform with time. Adaptive mesh refinement is also available.

- The transient analysis module (TR)

  This assumes the excitation current or voltage is of any form that is described by a variation over time. The solution is determined at discrete time steps. The transient waveform may be selected from standard driving functions or may be explicitly defined in tabular format.

- The velocity analysis module (VL)

  This assumes that specified regions are travelling at a specific velocity. Only static excitation is allowed. Adaptive mesh refinement is also available.

- The rotating machines module (RM)

  This assumes that the device is rotating, and hence inducing eddy currents. The excitation current or voltage is of any form that is described by a variation over time. The solution is determined at discrete time steps. The transient waveform may be selected from standard driving functions or may be explicitly defined in tabular format. The material properties can be non-linear (i.e. the permeability may be a function of the field strength at each point in the material).

- The thermal analysis module (TH)

  This carries out thermal analysis based on the solution of the electromagnetic analysis. It may also be used independently.

1. Only analysis modules that are licensed are available for use

- The transient thermal analysis module (THTR)

  This carries out thermal analysis as above, but for excitations that are of any form that are described by a variation in time. It may also be used independently.

- The stress analysis module (SA)

  This carries out stress analysis based on the solution of the electromagnetic analysis. It may also be used independently.

The problem to be modelled has a constant drive current and hence can be solved using the static analysis module. From the main menu select

FILE ↓
Write file → **Analysis data** → **Static analysis**

A menu listing all static analysis options appears. Accept all defaults by hitting **Return** twice.

Storing a problem

It is essential to store the model[1]. To save the model select

FILE ↓
Write file → **Write model**

In the file selection box enter **disc1** and press the **Accept** button. This creates two OPERA-2d files containing the model and mesh data. These are *disc1.op2* and *disc1.mesh* respectively. The *.op2* file contains all the model data and must be kept. The *.mesh* file is less important[2] and can be deleted if storage space is limited. It is useful as it avoids the need to re-mesh the model, which can be time-consuming for large models.

The Static Analysis module

In order to launch the Static Analysis module select

FILE ↓
Start analysis

1. This can be done at any time during the pre processing stage, and should be used regularly to back up your work while building the model
2. If an adaptive analysis was used - the mesh file *must* be kept as well as the *.op2* data file, as this mesh data cannot be re-produced from the model data

Within the **Start analysis** menu select the appropriate filename (*disc1*) and **Accept**.

OPERA-2d then reports on the progress of the solution in a text window.

On successful completion, OPERA-2d returns to the initial screen. The solution process has created a solution file, *disc1.st*. This contains the solution data from the static analysis for post processing.

It must be stressed that the analysis program will run from within the pre and post processing environment, provided enough system resources are available. Alternatively, the solver programs may still be accessed via the usual system route. For example, in Unix, to run the ST solver type **opera**[1] from the operating system prompt, followed by **ST.**

1. If this does not work, ask your system administrator to set up an alias for **opera**.

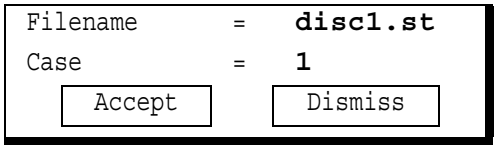
Post Processing

Examining the Solution

From the main menu select

FILE ↓
Read File → **Read model**

A file box is displayed from which the required file, *disc1.st* should be selected and the parameter box completed as shown below,



before selecting **Accept**

An information window is displayed showing the units setting. This may be cleared by hitting any key.

Select **Return** twice to return to the top level command menu.

Now select

DISPLAY ↓
Refresh

and the model regions are displayed. The display of the complete model is shown in Figure 4.21.

Line Contours of modified vector potential

The model solution potential was selected as R*A. With an axisymmetric solution, contour lines of constant r***A** represent flux lines[1]. To display the flux line (field) distribution, select:

FIELDS ↓
Contour plot → **Execute**

1. In XY symmetry using a vector potential, equipotential lines represent lines of magnetic flux

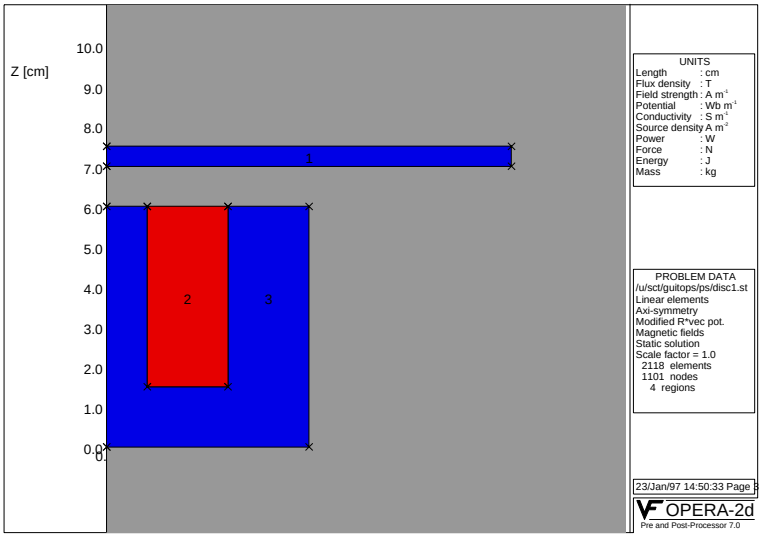


Figure 4.21 The complete model in the pre and post processor

An information window displaying the model error is given. This can be cleared by hitting any key. Vector equi-potential lines (which can be thought of as flux lines) are displayed over the whole model. The resulting display is shown in Figure 4.22.

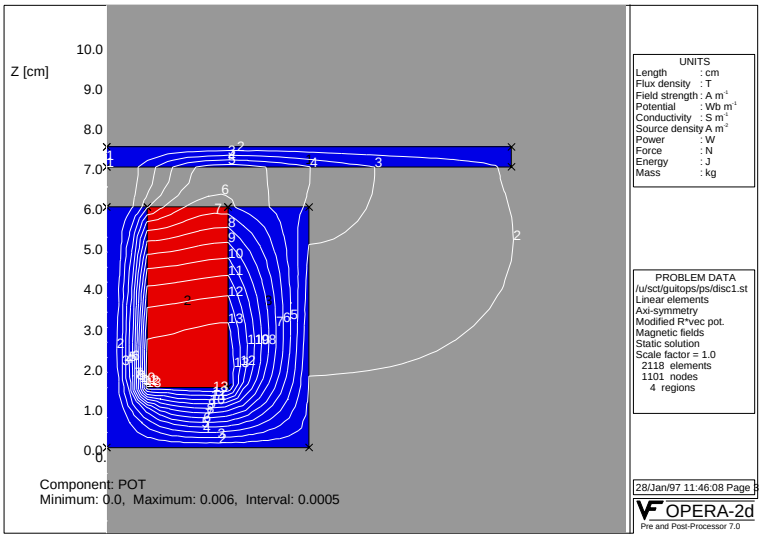


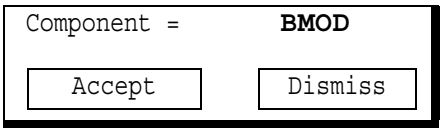
Figure 4.22 Equipotential contours of vector potential

Zone Contours of Flux Density Magnitude

In the previous contour plot, vector potential (component name POT) was displayed. To change this to the magnitude of flux density (component name BMOD) select

Fields ↓
Component

and complete the parameter box as shown below



It is possible to restrict the display to particular regions or material types. In this case the flux density display is to be restricted to regions of material 1, 3 and 4 (and not 0 i.e. air regions) by selecting the menus shown in Figure 4.23.

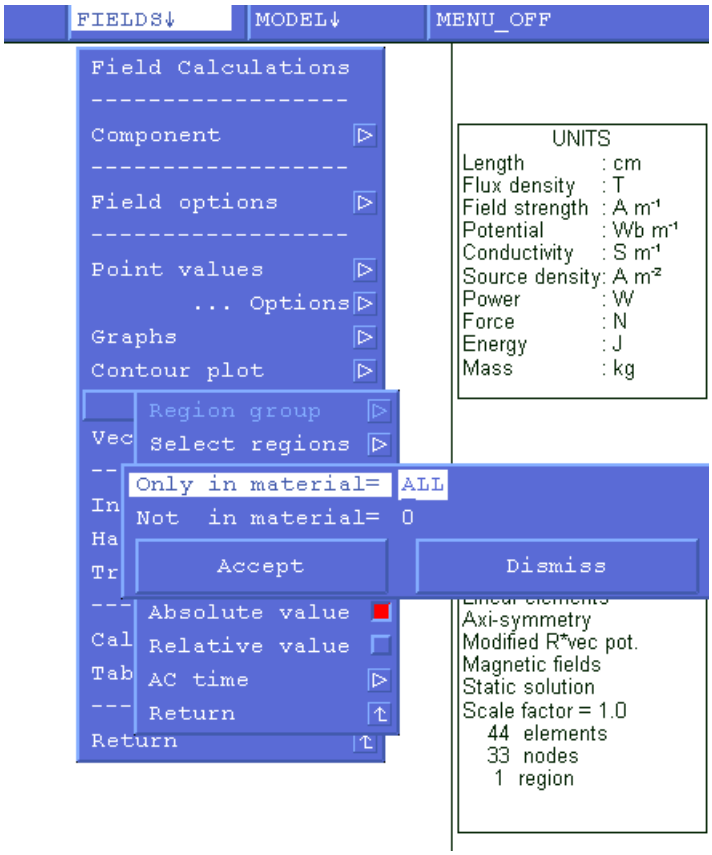


Figure 4.23 Menu selection for restricting contours to materials 1, 3 and 4

Fields ↓

.....Options *(under Contour plot)* → **Select material**

Only in material	=	**All**
Not in material	=	**0**
Accept		Dismiss

and **Return**. The contour style can be changed to zone contours. This is done by

Fields ↓

Contour plot → **Style** → **Filled zones**

and **Return**, followed by:

Fields ↓

Contour plot → **No refresh** *(toggles to refresh)*

Fields ↓

Contour plot → **Number of lines**

Number of lines	=	**30**
Accept		Dismiss

Contour plot → **Execute**

This gives zone contours of the modulus of flux density in the ferromagnetic regions, as shown in Figure 4.24.

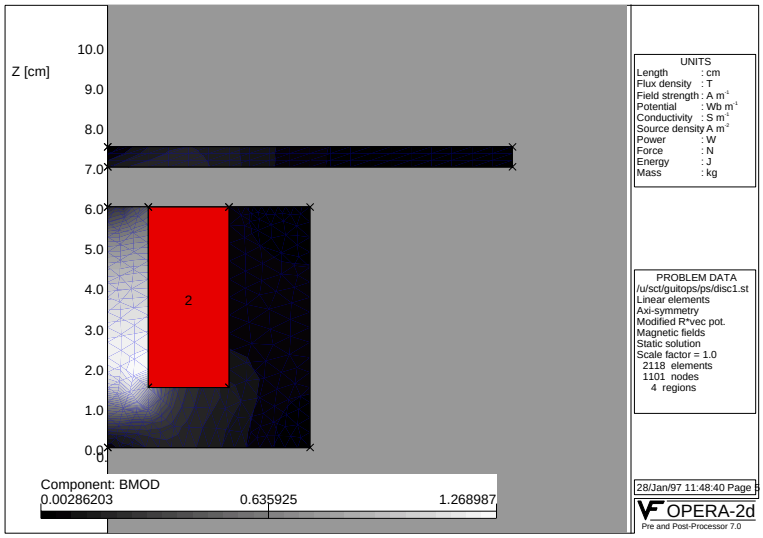


Figure 4.24 Zone Contours of Modulus of Flux Density

Graphs of Flux Density Component with Position

To examine the variation of flux density in the z-direction, along a line between the disc and the former, change the component to be that of flux density in the Z direction (component name BZ) by

FIELDS ↓
Component

Component =	**BZ**
Accept	Dismiss

To specify the required line

Graphs → **Along line**

and complete the parameter box with the data set to

Start X coordinate	=	**0**
......Y coordinate	=	**6.5**
End X coordinate	=	**10**
....Y coordinate	=	**6.5**
Curvature of line	=	**0**
Number of points	=	**200**
Accept		Dismiss

The line chosen is indicated on the model regions. This is followed by the graph of BZ against R between the disc and the former as shown in Figure 4.25.

Close the menus by selecting **Return** twice.

To leave the pre and post processor select

FILE ↓
End OPERA-2d/pp

and confirm this by selecting **Yes**

OPERA-2d returns to the initial screen. On Unix, to return to the operating system select

Quit

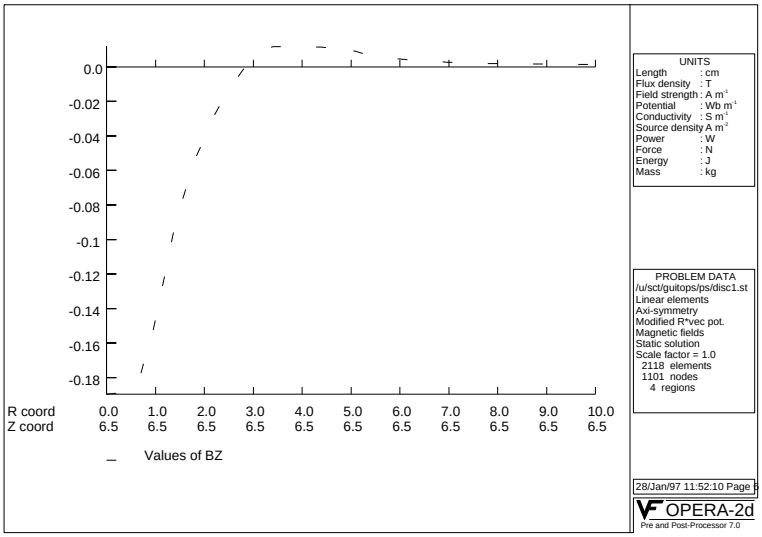


Figure 4.25 Graph of B_z against r

Chapter 5
Analysis and Utility Programs

Introduction

This manual describes the electromagnetic, stress and thermal analysis programs within the OPERA-2d environment. The analysis programs incorporate state of the art algorithms for the calculation of electromagnetic fields and advanced finite element numerical analysis procedures. A brief introduction to the use of finite elements is given in section “The Finite Element Method” on page 5-2 as an aid to application engineers who need to understand the limitations of the technique and evaluate the validity of their results. In sections “Codes of Practice” on page 5-7 and “Solution Errors” on page 5-7, some further discussions on good codes of practice, and techniques for reducing errors are presented.

Finite element discretisation forms the basis of the methods used in the analysis programs. This widely applicable technique for the solution of partial differential equations requires special enhancements to make it applicable to electromagnetic field calculations. Access to these features is supported by the OPERA-2d pre and post processor which provides facilities for the creation of finite element models, specification of complicated conductor geometry, definition of material characteristics including for example, non-linearity and anisotropy and graphical displays for examination of the data. Full details are given in the OPERA-2d Reference Manual.

The Finite Element Method

The Finite Element method is used to obtain solutions to partial differential or integral equations that cannot be solved by analytic methods. Partial differential and integral equations describe the spatial and temporal variation of a field either directly in terms of the field variable, for example the magnetic flux density **B**, but more often using a potential function that is related to the Field by a gradient (∇) or curl ($\nabla\times$) operation. The finite element method is generally applicable to any problem with any type of non-linearity. The method is based on division of the domain of the equation (volume of space in which the equation is satisfied) into small volumes (the finite elements). Within each finite element a simple polynomial is used to approximate the solution.

The concepts used in finite element analysis are independent of the number of space dimensions, however it is convenient to use a simple one space dimension problem in order to make the algebra straightforward and explanatory diagrams easy to view. Consider a Poisson type equation describing a potential function ϕ in one dimension:

$$\nabla \cdot \varepsilon \nabla \phi = \rho \qquad (5.1)$$

The potential function ϕ might be an electrostatic potential, in which case ρ would be a line charge density. In order to define ϕ, boundary conditions are required, these may be either assigned values of ϕ or its derivative $\frac{\partial \phi}{\partial x}$, for example:

$$\frac{\partial \phi}{\partial x} = 0 \qquad (5.2)$$

In all electromagnetic field examples it is essential that the potential is defined at one point in the domain at least, otherwise an infinite number of solutions could be generated by adding an arbitrary constant to the solution.

To solve equation 5.1 using a finite element method the domain is divided into line elements. A typical first order line element would have two nodes numbered e.g. 1 and 2. Within this element the potential ϕ will be approximated by a linear polynomial:

$$\phi(x) = a + bx \qquad (5.3)$$

The electrostatic potential ϕ will be continuous over the domain, although its derivatives may be discontinuous if the permittivity ε changes discontinuously. The finite element model should be capable of representing this behaviour and it is therefore convenient to characterize the polynomial shown in equation 5.3 by

the values of ϕ at the nodes of the element and use the same nodal value to characterize the polynomials in other elements that meet at the node. A further simplification is introduced by rewriting equation 5.3 in terms of nodal shape functions N_i defined such that:

$$\begin{aligned} N_i(x) &= 1; \quad x = x_i \\ N_i(x) &= 0; \quad x = x_j, \quad j \neq i \end{aligned} \tag{5.4}$$

where x_i is the x coordinate of node i etc. The shape functions have the same polynomial form as the ϕ approximation, and equation 5.3 can be written as:

$$\phi(x) = N_1(x)\phi_1 + N_2(x)\phi_2 \tag{5.5}$$

The shape functions N_i are usually expressed in terms of a local coordinate system in the element. This can be used to simplify the expressions and furthermore avoids problems of numerical rounding errors. Using the local coordinate system ξ the shape functions can be written as:

$$\begin{aligned} N_1 &= \frac{1}{2}(1-\xi) \\ N_2 &= \frac{1}{2}(1+\xi) \\ -1 &\leq \xi \leq 1 \end{aligned} \tag{5.6}$$

The shape function for a particular node is only defined in the elements that use the node and is zero outside these elements. The approximation to ϕ is described as having local support when nodal shape functions of this type are used.

The discrete method of approximating the potential ϕ using characteristic nodal values and associated shape functions that determine the spatial variation of the approximation provides the basis on which several alternative procedures could be used to solve equation 5.1. Variational methods, least squares and weighted residual procedures are three of the most frequently used. Weighted residuals have wide application and they are used in the software to develop a numerical solution. An approximate solution ϕ is determined by requesting that this function should satisfy:

$$\int W(\nabla \cdot \varepsilon \nabla \phi - \rho)\,dx = 0 \tag{5.7}$$

The weighted residual method can be used with either global (defined over the whole domain) approximations to ϕ or the local approximations discussed here. W is a weighting function from which the method gains its name. The Galerkin weighted residual method is the best choice for the types of equation arising in electromagnetism. In this case the basis functions approximating ϕ are also used

for the weights. Equation 5.7 is often referred to as a strong form because of the constraints it places on the functions that can be used in the approximation ϕ (the first derivative would clearly have to be continuous over the domain). In general a weak form of equation 5.7 is used to remove the derivative continuity requirement. This weak form is obtained by integrating equation 5.7 by parts (in more than one dimension this involves application of Green's theorem). Integrating equation 5.7 by parts to reduce the order of differentiation applied to ϕ gives:

$$\int_a^b (\nabla W \cdot \varepsilon \nabla \phi + W\rho)\mathrm{d}x - \left[W\varepsilon\frac{\partial\phi}{\partial x}\right]_a^b \tag{5.8}$$

where a and b are the limits of the domain of the equation. The weak form has several advantages: the functions representing W and ϕ do not need derivative continuity and the natural boundary condition on the surface of the domain $\frac{\partial\phi}{\partial x}$ has emerged. Equation 5.8 leads directly to a numerical solution method, using the discrete finite elements and shape functions discussed above. Discretisation of the domain *ab* into line elements with their associated nodes gives a set of independent weighting functions (the shape functions of the nodes) from which a set of equations can be developed by requiring that equation 5.8 is satisfied independently for each weight function. The equation for weight function W_i, i.e. shape function N_i, is obtained from:

$$\sum_j \left(\int_a^b (\nabla N_i \cdot \varepsilon \nabla N_j \phi_j + N_i \rho)\partial x \right) - \left[N_i \varepsilon \frac{\partial\phi}{\partial x}\right]_a^b = 0 \tag{5.9}$$

for all elements containing node *i*. Taking all the equations for the different weight functions together gives a set of linear equations, which written in matrix form are:

$$\mathbf{K\Phi} = \mathbf{S} \tag{5.10}$$

where **K** is a coefficient matrix (often called a stiffness matrix because of the background of finite elements in mechanics), **Φ** is a vector of unknown nodal potential values and **S** the known right hand side vector derived from the prescribed line charge densities or assigned boundary conditions. An individual element of the stiffness matrix consists of terms of the form:

$$K_{\mathrm{ij}} = \int_a^b \nabla N_i \cdot \varepsilon \nabla N_j \mathrm{d}x \tag{5.11}$$

Note that the local support of the shape functions means that although the integral in equation 5.11 is taken over the whole domain, only elements containing both

nodes *i* and *j* actually contribute. In the equations arising in electromagnetism the matrix equation 5.10 is frequently non-linear because the value of ε (or more frequently μ for magnetic fields) is dependent on the field intensity.

Non-Linear Materials

A Newton-Raphson method can be used to solve this type of non-linear equation. Given an initial solution Φ_n a new solution Φ_{n+1} is found by solving the linearized Jacobean system[1]:

$$\Phi_{n+1} = \Phi_n - J_n^{-1} R_n \tag{5.12}$$

where the residual R is given by

$$R_n = K_n \Phi_n - S_n \tag{5.13}$$

and the Jacobean J by

$$J_n = \frac{\partial}{\partial \Phi_n} (K_n \Phi_n - S_n) \tag{5.14}$$

This method converges providing the initial approximation used to start the iteration is not too far from the real solution. As it approaches the solution its convergence becomes quadratic. In the context of non-linear finite element solutions to the electromagnetic field equations the reliability of the Newton-Raphson method is strongly linked to the smoothness of the equation used to relate the permeability or permittivity to the field.

To aid convergence, when the material property curve is not smooth or when the initial solution is far from the final solution, a relaxation factor is used in equation 5.12, which becomes:

$$\Phi_{n+1} = \Phi_n - \alpha J_n^{-1} R_n \tag{5.15}$$

where α is chosen, starting with 1 and multiplying by 2 (if the change in $|R|$ is too small) or dividing by 2 (if the norm of the residual $|R_{n+1}|$ would be greater than $|R_n|$) to find the value which minimises the norm of the residual $|R_{n+1}|$ at the start of the next iteration.

1. The subscript ‘n’ indicates the iteration number

Finite Element Applications

The experience of the user has been a vital ingredient in the successful application of finite elements to predictive engineering design. Large finite element systems for mechanical design have a mystique associated with them, partly from their origins as stand alone programs with a text file user input interface and a box of paper as the output display, and partly as a result of the jargon that is used to describe the element types and procedures available within the programs. It is now impossible to consider the use of finite element analysis programs without interactive pre processors for data input, although these bring their own problems if they do not interface well with the analysis programs and thus increase the amount of knowledge needed to perform a calculation.

In electromagnetic field calculations special finite elements are not needed to solve the equivalent of shell and plate geometries that are so common in mechanical design. However, electromagnetic fields must usually be computed to much higher accuracy than is needed in other disciplines, the geometry is frequently complicated with a wide range of dimensions and the actual result needed by the user is often derived from the field solution by integration or differentiation. In whatever form the results are required, the basic limitation of finite element solutions is that the accuracy of the solution is related to the size of the discrete elements. Recent research has resulted in the development of techniques that can be used to determine the error in a finite element solution but this is strictly only correct for problems with linear materials. The adoption of these techniques in finite element programs will improve the reliability of results, but they do not help to check that the finite element model and the physical problem are equivalent.

In the pre and post processor, the local error (in flux density units) and the *rms* error over the whole problem can be accessed via the system variables ERROR and RMSERROR. The value of *rms* error is displayed by the program in two ways:

- ‘RMS error over whole problem’ is the value stored in the system variable RMSERROR. It is calculated from the relative errors in each element.
- ‘Weighted RMS error’ uses the element sizes as weights in the calculation so that smaller elements make a smaller contribution to the value. A different method is used in the adaptive analysis routines.

It should be remembered that these values are an upper-bound on the real errors and also that a poor discretisation will not only give poor answers, but also poor estimates for the errors.

Codes of Practice

Users of finite element programs must prove that the model is consistent with the physical problem. With electromagnetic fields it is often possible to perform simple calculations that give *orders of magnitude* answers as an essential part of the analysis. Until the accuracy of the model has been established it is irrelevant to consider the discretisation errors. A code of practice should be established that is followed whenever a new analysis is begun:

1. Solve the simplest possible problem first i.e. using linear materials either with unity or large relative permeability or permittivity.
2. Check that the solution has the symmetry that is expected. For example, examine the fields on the boundaries of the problem to see if they are as expected, with the proviso that on boundaries where the normal derivative condition is weakly imposed the derivative will not be exactly as defined (see section on boundary conditions later).
3. Check that the solution agrees with simple line integral predictions or images if infinite permeability approximations are applicable.
4. If the field is varying with time, check that the skin depth agrees with limiting case calculations or that the time constants agree with equivalent circuit models.

Only when a degree of confidence has been established in the model is it worth beginning to consider the errors produced by the discrete finite element approximation. In many ways these errors are more straightforward to evaluate than the accuracy of the model.

Solution Errors

The local error at a point within a finite element model is strongly linked to the size of the elements surrounding the point and weakly linked to the average element size over the whole space, although this second source of error becomes more important and less easily estimated in non-linear solutions. The relationship between the local error in the solution and the surrounding elements' size is given by:

For linear shape functions

$$E(\Phi) = O(h^2) \tag{5.16}$$

and for quadratic shape functions

$$E(\Phi) = O(h^3) \tag{5.17}$$

where E is the error, O means 'of the order' and h is the linear dimension of the elements. This simple analysis is only true for square elements, but it is reasonable to assume the worst case and use the largest dimension for h. Unfortunately, these formulae only give the order of the error, the actual error is dependent on the solution, or more precisely the geometry of the model in the vicinity of the point. As an example consider a point close to the corner of a magnetised steel cube, the field will be weakly singular at the corner. Given the same size discretisation over the whole space, the errors will be far larger close to the edges and corners of the cube. This is because the low order polynomials used in the finite elements are not good at approximating the singularity. Calculating the magnetic field from the potential solution generally results in larger errors in the field than there were in the potential.

Differentiation of the finite element shape functions to determine the field gives an error in the field that is worse by $O(h^{-1})$. In the case of linear shape functions this results in an error in the field $O(h)$. In the analysis modules special facilities have been included in order to reduce the errors in the fields that are computed from potential solutions. Two methods are available that increase the field precision; the best method depends on the problem being solved (see sections on accuracy in the following chapters). Nodal weighted averaging improves the field accuracy to $O(h^2)$. The volume integration technique does not improve the order of the error, but it enables the variation of the field to be calculated very accurately remote from magnetic, dielectric or conducting regions.

The programs use error estimation techniques to produce local and global errors in the fields derived from potential solutions (system variables ERROR and RMSERROR). These displays show where the finite element discretisation needs refinement as well as showing the error. However, even with these features, it is important that the user of a finite element program carries out a number of analyses to examine the effect of element size on the solution. Using the ideas introduced above it is clear that the best approach is to solve the same model with two levels of finite element discretisation or with the same discretisation but using linear elements in one case and quadratic elements in the other. Taking as an example the use of two levels of discretisation, such that the element dimensions are halved in the second case, the case with the larger number of elements will have solution errors that are 4 times smaller (the errors in the fields evaluated by differentiation of the shape functions will be halved). Examination of the changes between the two solutions will give a good estimate of the discretisation errors, but not, as pointed out in the previous section, any indication of the accuracy of the model.

When increasing the overall discretisation becomes too expensive it is necessary to carry out more trial analyses, in each case choosing particular regions of increased discretisation to determine the sensitivity of the solution to the change

in element size. An experienced user will have learnt how to minimise the number of trials as a result of carrying out this type of experiment on a number of different geometries. Unfortunately a little knowledge may be a dangerous thing! Even experts cannot rely completely on their past experience. Results must always be critically examined on the assumption that they are incorrect.

The adoption of error analysis techniques in finite element programs does not reduce the user's responsibility for the quality of the results.

Application Notes

This section describes how certain features of the programs should be used to achieve the best possible solutions to certain classes of problems.

Boundary Conditions

Boundary conditions are used in two ways. Firstly they can provide a way of reducing the size of the finite element representation of symmetrical problems. Secondly they are used to approximate the magnetic field at large distances from the problem (far-field boundaries).

General

Problem symmetry and the symmetry of the fields are implied by the potential boundary conditions applied to the finite element model. The simplest types of boundary condition are:

SIMPLE BOUNDARY CONDITIONS		
Field Symmetry	**Scalar Potential**	**Vector Potential**
$\mathbf{H}_n = 0$ or $\mathbf{B}_n = 0$	$\frac{\partial \phi}{\partial n} = 0$	$\mathbf{A}$=*constant*
$\mathbf{H}_t = 0$ or $\mathbf{B}_t = 0$	ϕ=*constant*	$\frac{\partial \mathbf{A}}{\partial n} = 0$

where the *n* and *t* subscripts refer to the normal and tangential directions to the surface being considered.

In electrostatic fields electrode surfaces will obviously have assigned potential boundary conditions (*v=value*). Except for this case, the other boundary conditions shown in the table above should only be applied to the exterior surfaces of the finite element model. The default condition that will always be applied if no boundary condition is specified on a exterior surface is:

DEFAULT BOUNDARY CONDITION		
Field Symmetry	**Scalar Potential**	**Vector Potential**
$\mathbf{H}_n = 0$	$\frac{\partial \phi}{\partial n} = 0$	
$\mathbf{B}_t = 0$		$\frac{\partial \mathbf{A}}{\partial n} = 0$

The default boundary condition is only weakly satisfied even when it is assigned to a surface. ‘Weakly satisfied’ means that it is applied as an integral over the surface patch for each shape function sub-domain. Examination of the field solution at, or close to, a surface with the default boundary condition will reveal that the normal component is not zero. The magnitude of the normal component can reflect the local accuracy of the solution or it may be a result of a model error.

The fields computed by taking derivatives of the finite element shape functions will be discontinuous, but in each element the potential boundary condition will be exactly specified. The field averaging processes in the OPERA-2d pre and post processor cannot model these discontinuities and this results in the field not exactly matching the boundary conditions.

Uniqueness

A potential boundary condition must be specified on at least one node of a finite element mesh. This gauges the potential and without it the solution will not be unique. Any constant value could be added over the whole space. In OPERA-2d this condition is satisfied by one surface in the problem having an assigned potential.

Open Boundaries

Electromagnetic fields are frequently not contained within a finite volume. In practice, at long distances from the device producing the field, the distribution will be modified by the local environment, but this will not effect the field close to the device. Except where the interaction with the far environment is of interest, the field from an isolated device is usually required. A simple finite element mesh obviously has a finite extent and applying either potential or derivative boundary conditions on the open boundaries will perturb the true infinite domain solution.

There are techniques that can be used to accurately model the infinite domain. On a convex outer surface a series of rings of elements with increasing size may be recursively generated automatically (ballooning) or a boundary integral solution for the exterior domain may be coupled to the interior finite element solution. The simplest to apply in OPERA-2d is the Kelvin transformation (or $\frac{1}{r}$ mapping).

Approximate methods include matching a convex outer surface to finite elements that extend from the surface to infinity and which have appropriate decay functions, and the standard approach of extending the finite element mesh to a distance where the field truncation has no effect on the regions of interest.

It is recommended that the Kelvin transformation be used, or that the mesh be extended to a reasonable distance with either potential or derivative boundary conditions applied to the outer surface, so that the truncation has an insignificant effect on the region of interest. The effect of truncation can be estimated by observing the non-zero component of the field on the open boundary. Half the field observed on such a boundary is being reflected back from the exterior. Combining this with knowledge of the probable decay in the exterior space will give an order of magnitude for the effect of the truncation on the regions of interest. In particularly sensitive applications the Kelvin transformation should be used or a further test should be applied to check the effect of truncation. Two problems should be solved, one using potential boundary conditions on the open boundary, the other using derivative conditions. These problems represent the model in an infinite array of similar problems with either the opposite or the same sign of field in alternate images. In general the two solutions will bound the correct answer. It is often found that the condition which gives zero normal field is closer to the real solution. It would in fact be exact if the open boundary surface was a constant flux surface. Use of this simple approach gives an estimate for the effect of the far field boundary truncation and this is why it is recommended.

Periodicity Conditions

The potential and derivative boundary conditions discussed in the previous sections are the most common in nearly all applications. There is one other class of boundary condition that frequently occurs in electrical machine design. When the problem has rotational symmetry about an axis or displacement symmetry, without any reflection symmetry in the same symmetry group, it is not possible to identify surfaces where the field is normal or tangential. The complete model could be defined but this would be unnecessarily expensive. In problems with this type of symmetry, pairs of matching surfaces can be identified where the potentials have the same sign and magnitude (positive) or the reverse sign but equal magnitude (negative).

The Basic Field Equations used in OPERA-2d

The formulations used in the analysis programs are based on the low frequency limit of Maxwell's Equations. They are listed here and are referred to by the following sections.

Magnetostatics and Eddy Currents

The magnetic field strength (**H**) and the current density (**J**) are linked by

$$\nabla \times \mathbf{H} = \mathbf{J} \tag{5.18}$$

and the electric field strength (**E**) and the magnetic flux density (**B**) by

$$\nabla \times \mathbf{E} = -\frac{\partial \mathbf{B}}{\partial t} \tag{5.19}$$

B has zero divergence:

$$\nabla \cdot \mathbf{B} = 0 \tag{5.20}$$

B can be expressed in terms of **H** and the permeability (μ),

$$\mathbf{B} = \mu(\mathbf{H} - \mathbf{H}_c) \tag{5.21}$$

where $\mathbf{H}_c$ is the coercive force of any permanent magnetic material. **J** can be expressed in terms of **E** and the conductivity (σ)

$$\mathbf{J} = \sigma\mathbf{E} \tag{5.22}$$

It is convenient to define a magnetic vector potential, **A**, by

$$\mathbf{B} = \nabla \times \mathbf{A} \tag{5.23}$$

and a magnetic scalar potential, ϕ by

$$\mathbf{H} = -\nabla\phi \tag{5.24}$$

Electrostatics

The displacement current (**D**) is related to the charge density (ρ) by

$$\nabla \cdot \mathbf{D} = \rho \tag{5.25}$$

and to the electric field strength and permittivity (ε) by

$$\mathbf{D} = \varepsilon\mathbf{E} \tag{5.26}$$

The electric scalar potential (V) is defined by

$$\mathbf{E} = -\nabla V \tag{5.27}$$

Static Field Analysis (ST)

The Static Field Analysis Program (OPERA-2d/ST) solves for time invariant magnetic or electric fields. The model can include non-linear material permeability or permittivity and either infinite XY or axisymmetric coordinate systems. Other types of field that are described by a non-linear Poisson equation can also be solved, these include current flow, Newtonian flow and static temperature fields.

The Equations Solved

OPERA-2d/ST solves for the vector or scalar potential defined by a non-linear Poisson equation. The vector potential form is usually used for magnetic field analysis because the scalar potential solution cannot include current as the source of the fields, however if a model is only excited by boundary conditions or by permanent magnets then either form can be used. The equation to be solved for the static magnetic field case using a magnetic vector potential is derived by substituting $\frac{1}{\mu}\nabla\times\mathbf{A}$ for $\mathbf{H}$ in equation (5.18), this gives

$$\nabla\times\left(\frac{1}{\mu}\nabla\times\mathbf{A}-\mathbf{H}_c\right) = \mathbf{J} \tag{5.28}$$

The alternative scalar potential form is derived by substituting $\mu\nabla\phi$ in equation (5.20) to give

$$\nabla\cdot\mu(\nabla\phi-\mathbf{H}_c) = \rho \tag{5.29}$$

The value of ρ must be zero for a magnetic scalar potential, however if ϕ is an electric scalar potential then ρ is the charge density. The term involving $\mathbf{H}_c$ represents permanent magnets for magnetic field analysis and electrets for electric field analysis.

OPERA-2d offers a number of choices of solution potential that can be selected by using the SET command. The recommended choice for a number of applications is

Recommended Static Field Solution Choices		
Field Type	**Symmetry**	**SET Solution type**
MAGNETIC	XY	**SOLUTION=AT** (Vector Potential)
MAGNETIC	AXI	**SOLUTION=AXI** (Modified rA) (best choice)
MAGNETIC	AXI	**SOLUTION=AT** (Vector Potential) (second best)
ELECTRIC	XY	**SOLUTION=V** (Scalar Potential)
ELECTRIC	AXI	**SOLUTION=V** (Scalar Potential)

Data interpretation

The OPERA-2d/ST solution program expects material code number 0 to be free space with zero density, material number 1 to have density not equal to zero and relative permeability (permittivity) of one. Material code numbers greater than 2 must have BH (DE) characteristics defined, even for linear solutions where the region value of PERM is used and the density parameter may be non zero. BH characteristics are required because the value of **H** for the first point of the table (which must have **B** equal zero) defines the $\mathbf{H}_c$ for the material.

The region parameter PHASE defines the easy direction of magnetisation for a permanent magnet material. The remanent magnetisation is the negative X direction of a local coordinate system that is rotated anticlockwise by an angle PHASE in degrees from the X axis.

Preparing an OPERA-2d/ST run

The data required to analyse the model is input using the SOLVE command (see Reference Manual for details). The user can set the following information:

- Linear or Non-linear solution

  The linear solution options simply solves with the value of permeability (permittivity) given by the region parameter PERM. A BH curve is still required to give a value for the coercive force. The Non-linear option solves for the non-linear materials as defined by the BH (DE) characteristics defined for material numbers greater than 2. If Non-Linear is chosen the following parameters are set:

  - Number of iterations can be set if a non-linear solution is requested. The value entered sets a limit on the maximum number of non-linear iterations.
  - Iteration type (Newton or simple - the simple iteration method is only provided as a fall-back if Newton iterations fail).
  - Tolerance is the convergence tolerance to be applied to the relative change in the solution.

- ‘Mesh refinement’ or ‘No mesh refinement’

  This defines whether the mesh is allowed to automatically refine during the analysis. If mesh refinement is selected, the following can be set:

  - Maximum number of iterations
  - Maximum number of elements
  - Final convergence accuracy %

- Scale factor

  The scale factor multiplies charge or current densities and non-zero boundary condition values. A list of scaling factors can be supplied.

- ‘New solution’ or Restart

  This is only available if the *.op2* file already contains a solution.

Steady-state AC Analysis (AC)

The Steady-state ac Analysis Program (OPERA-2d/AC) solves eddy current problems where the driving currents or voltages are varying sinusoidally in time. It can analyse skin effect, quasi-non-linear materials, in both XY and axisymmetric coordinate systems.

The Equations Solved

OPERA-2d/AC solves the vector Helmholtz equation with the magnetic vector potential (**SET SOLUTION=AT**) as the unknown variable. In axisymmetry better answers can be obtained using the modified rA potential (**SET SOLUTION=AXI**).

The equation to be solved is formed from equations (5.18) and (5.21) (without the coercive force, $\mathbf{H}_c$), (5.22) and (5.23), and the integration of (5.19):

$$\nabla \times \left(\frac{1}{\mu} \nabla \times \mathbf{A} \right) = \mathbf{J}_s + \mathbf{J}_v - \sigma \frac{\partial \mathbf{A}}{\partial t} \tag{5.30}$$

in which the current density has been split into the prescribed sources, $\mathbf{J}_s$, the currents in windings of external circuits, $\mathbf{J}_v$, and the induced currents, $\sigma \frac{\partial \mathbf{A}}{\partial t}$. In two dimensions, only the *z* components of **A**, $\mathbf{J}_s$ and $\mathbf{J}_v$ exist.

Equation (5.30) can be simplified to:

$$-\nabla \cdot \frac{1}{\mu} \nabla A_z = J_s + J_v - \sigma \frac{\partial A_z}{\partial t} \tag{5.31}$$

Since the potential and the currents are varying sinusoidally, they can be expressed as the real parts of complex functions $A_c e^{j\omega t} t$ and $J_c e^{j\omega t}$. Equation (5.31) now becomes

$$-\nabla \cdot \frac{1}{\mu} \nabla A_c = J_c + J_v - i\omega\sigma A_c \tag{5.32}$$

which is solved using complex arithmetic.

Eddy Current Conductors

In some problems it is necessary to allow eddy currents to flow in driving conductors, thus reducing the total current transported by the conductor. In other cases the total current in the conductor is prescribed and therefore the total eddy current in the conductor must be zero. This second case is achieved by solving an extra equation for each group of regions which make up such a conductor. This equation limits the total current, *I*, flowing in terms of $\frac{\partial A}{\partial t}$ and a potential gradient, ∇V, which comes from the integration of equation (5.19) and is usually zero.

$$-\int_{\Omega_I} \sigma\left(\frac{\partial A}{\partial t} + \nabla V\right)\partial\Omega = I \tag{5.33}$$

The effect of the potential gradient is a spatially uniform current density over the conductor in xy symmetry and varying as $\frac{1}{r}$ in axisymmetry. In xy symmetry, J^* is given by

$$J^* = -\sigma\nabla V \tag{5.34}$$

It becomes an extra unknown in the modified equation (5.31). The following two equations are solved together

$$\nabla \cdot \frac{1}{\mu}\nabla A_c = J_c^* - i\omega\sigma A_c \tag{5.35}$$

$$\int_{\Omega_J} (-i\omega\sigma A_c + J_c^*)\partial\Omega = \int_{\Omega_J} J_c \partial\Omega \tag{5.36}$$

Equation (5.36) is repeated for every group of regions with a different value of N and SYMMETRY≠0. J_c is given by DENSITY and PHASE.

External circuits

Steady-state ac solutions can be excited either by current sources or by voltage sources connected via external circuits to the model. The current density that can be defined for each region is supplied by a current source, independent of the properties of the coils the current is given as a function of time. The voltage driven option allows a set of coils to be defined as a circuit which is connected to an external voltage source in series with an external resistance, capacitance and inductance. A coil is the set of regions that have the same conductor number N.

A full description of the external circuit option is provided in the Reference Manual. Hints on the use of external circuits are also given in a separate Application Note.

Permeability

The solution method in OPERA-2d/AC uses permeabilities in the following manner:

(a) In a linear solution, the permeabilities are constant over each region, and defined in the region data

(b) In a new quasi non-linear solution, element values are calculated from the maximum field in the ac cycle. This option is an approximation that can be useful in some applications, but it must be used with care

(c) Finally, in a restart solution, element values of permeability obtained from a previous solution (for example calculated from non-linear statics or transient analysis) are used. A 'restart' option is offered which can be used to restart an approximate (maximum field) non-linear steady-state solution or to continue from a previous static or transient solution.

If the restart option is selected and linear materials are requested then the permeability can be calculated in two ways: the absolute (DC) values of permeability or the incremental (AC) values of permeability ($\frac{\partial \mathbf{B}}{\partial \mathbf{H}}$) at the field level of the restart solution can be calculated from the BH curve. This second option allows solution of problems where the ac field is a small perturbation on a background field. The perturbation and background fields can be combined in the post processor with the APPEND command.

In all instances, the OPERA-2d/AC solver also offers the option of permeabilities being complex. This simulates hysteresis by introducing a phase lag between **H** and **B**. This lag is specified with the BH curve.

Boundary Conditions

OPERA-2d/AC supports three types of boundary condition: fixed potential (**F*n*=V, V*n*=*value***), fixed derivative (**F*n*=DV, DV*n*=0**) and periodic boundaries (**F*n*=SYMMETRY**). All other boundaries should be set to **F*n*=NO**, which is equivalent to **F*n*=DV, DV*n*=0**.

Preparing OPERA-2d/AC problems

The data required to analyse the model is input using the SOLVE command (see Reference Manual for details). The user can set the following information:

- Linear or Non-linear solution

  The linear solution options simply solves with the value of permeability (permittivity) given by the region parameter PERM. The Non-linear option solves for the non-linear materials as defined by the BH (DE) characteristics defined for material numbers greater than 2. If Non-Linear is chosen the following parameters are set:

  – Number of iterations can be set if a non-linear solution is requested. The value entered sets a limit on the maximum number of non-linear iterations.

  – Tolerance is the convergence tolerance to be applied to the relative change in the solution.

- 'Mesh refinement' or 'No mesh refinement'

  This defines whether the mesh is allowed to automatically refine during the analysis. If mesh refinement is selected, the following can be set:

  – Maximum number of iterations

  – Maximum number of elements

  – Final convergence accuracy %

- Frequency

  A list of frequency values may be supplied.

- Complex permeability on or off

  Complex permeability is used to define a lossy material.

- 'New solution' or Restart

  This is only available if the *.op2* file already contains a solution. The AC solution program allows a solution to be continued from the result contained in an existing results file created by one of the non-linear analysis programs (ST or TR) in order to pick up the element permeabilities. Alternatively, a run may be restarted from an incompletely converged AC non-linear run.

  For restart runs only, the 'Static' and 'incremental' permeabilities from the previous non-linear analysis may be used. 'Static' and 'incremental' permeabilities are described on page 5-20.

Transient Analysis (TR)

The Transient Analysis Program (OPERA-2d/TR) solves eddy current problems where the driving currents or voltages are changing in time in a predetermined way. It can analyse the response to multiple drive functions including a dc background field, skin effect, non-linear materials, XY and axisymmetric coordinate systems.

The Equations Solved

OPERA-2d/TR solves the vector diffusion equation with the magnetic vector potential (**`SET SOLUTION=AT`**) as the unknown variable. It is formed from equations (5.18), (5.21), (5.22) and (5.23), and the integration of (5.19):

$$\nabla \times \left(\frac{1}{\mu} \nabla \times \mathbf{A} - \mathbf{H}_c \right) = J_s - \sigma \frac{\partial \mathbf{A}}{\partial t} \qquad (5.37)$$

in which the current density has been split into the prescribed sources, $\mathbf{J}_s$ and the induced currents, $\sigma \frac{\partial \mathbf{A}}{\partial t}$. In two dimensions, only the z components of $\mathbf{A}$ and $\mathbf{J}_s$ exist. Equation (5.37) can be simplified to:

$$-\nabla \cdot \frac{1}{\mu} \nabla A_z - (\nabla \times \mathbf{H}_c)_z = J_s - \sigma \frac{\partial A_z}{\partial t} \qquad (5.38)$$

Driving Functions

The ‘driving’ field is provided by source currents, J_s, voltages, non-zero potential boundary conditions, and permanent magnet coercive forces, $\mathbf{H}_c$. The permanent magnets obviously create a dc field. The other driving terms can have a prescribed shape in time. Each value of conductor number (region parameter N) can be related to a different driving function, with, if necessary, every region having a different function.

The driving function options are:

Transient Driving Functions	
Name	**Function**
COSINE	$t<0$: $F = 1$ $t\geq 0$: $F = \cos(2\pi ft - \phi)$. The program prompts for f and ϕ.
DC	Uniform in time from $t=-\infty$ to $t=+\infty$
PEAK	$t<0$: $F = 0$ $t \geq 0: F = at\exp\left(\frac{-t^2}{b}\right)$. The program prompts for t_c. a and b are chosen such that $F=1$ at $t=t_c$.
RAMP	$t<0$: $F = 0$ $t \geq 0, t \leq t_c: F = \frac{t}{t_c}$ $t>t_c$: $F = 1$. The program prompts for t_c.
EXPONENTIAL	$t<0$: $F = 0$ $t \geq 0: F = 1 - \exp\left(\frac{-t}{t_c}\right)$. The program prompts for t_c.
SINE	$t<0$: $F = 0$ $t\geq 0$: $F = \sin(2\pi ft - \phi)$. The program prompts for f and ϕ.
STEP	$t<0$: $F = 0$ $t\geq 0$: $F = 1$
TTOFF	$t<0$: $F = F(0)$ $t\geq 0$: $F =$ cubic splines The program prompts for the name of a time-table file. F(0) is the value of the function in the table file at time t=0.
TTON	$t<0$: $F = 0$ $t\geq 0$: $F =$ cubic spline The program prompts for the name of a time-table file.

The transient time table option allows the user to define driving functions other than those programmed into the analysis code. The tables consist of files containing up to 1000 pairs of numbers in free format, one pair per line. The first number on each line specifies the time; the second gives the function value. The values of

time should start at zero and increase through the file. Discontinuities in function value or first derivative can be forced by specifying two entries for the same value of time. For switch-on cases, the function value at zero time need not be zero, but it is assumed that the function has value zero for all time before zero. Beyond the last value of time in the table the function continues with the same cubic function calculated for the last section of the table.

The driving functions may be viewed using the **Graphs** Option, available from the **FILE** Top Level Menu.

Eddy Current Conductors

In some problems it is necessary to allow eddy currents to flow in driving conductors, redistributing the current density, or to limit the total eddy current in a conductor to zero. This is achieved by solving an extra equation for each group of regions which make up such a conductor. This equation limits the total current, I, flowing in terms of $\frac{\partial A}{\partial t}$ and a potential gradient, ∇V, which comes from the integration of equation (5.19) on page 5-13 and is usually zero.

$$-\int_{\Omega_I} \sigma\left(\frac{\partial A}{\partial t} + \nabla V\right)\partial\Omega = I \tag{5.39}$$

The effect of the potential gradient is a spatially uniform current density over the conductor, J^* given by

$$J^* = -\sigma\nabla V \tag{5.40}$$

and becomes an extra unknown in the modified equation (5.38). The following two equations are solved together

$$-\nabla\cdot\frac{1}{\mu}\nabla A_z - \nabla\times\mathbf{H}_c = J^* - \sigma\frac{\partial A_z}{\partial t} \tag{5.41}$$

$$\int_{\Omega_J}\left(-\sigma\frac{\partial A}{\partial t} + J^*\right)\partial\Omega = \int_{\Omega_J} J_s\partial\Omega \tag{5.42}$$

Equation (5.42) is repeated for every group of regions with a different value of N and **SYMMETRY≠0**. J_s is given by DENSITY.

External circuits

Transient solutions can be excited either by current sources or by voltage sources connected via external circuits to the model. The current density that can be defined for each region is supplied by a current source, independent of the properties of the coils the current is given as a function of time. The voltage driven option allows a set of coils to be defined as a circuit which is connected to an external voltage source in series with an external resistance, capacitance and inductance. A coil is the set of regions that have the same conductor number N.

The Reference Manual provides a full description of External Circuits options. Hints on the usage of external circuits are given in a separate Application Note.

Time Stepping

The finite element method used in OPERA-2d/TR is similar to that described previously. However the terms in $\boldsymbol{A}$ and $\frac{\partial A}{\partial t}$ in equations (5.38) both yield matrices, referred to as **R** and **S**. The Galerkin procedure leads to a matrix equation

$$\boldsymbol{R}\boldsymbol{A} + \boldsymbol{S}\frac{\partial \boldsymbol{A}}{\partial t} + \boldsymbol{B} = 0 \qquad (5.43)$$

where $\boldsymbol{A}$ is now a vector of unknown potentials and $\boldsymbol{B}$ is a vector of driving terms. The solution of equation (5.43) is also based on the Galerkin procedure. $\boldsymbol{A}$ and $\boldsymbol{B}$ are discretized in time using a first order function of time:

$$\boldsymbol{A}(t) = (1-\tau)a_n + \tau a_{n+1} \qquad (5.44)$$

$$\boldsymbol{B}(t) = (1-\tau)b_n + \tau b_{n+1} \qquad (5.45)$$

where

$$\tau = \frac{t - t_n}{t_{n+1} - t_n} \qquad (5.46)$$

and a_n and b_n are values of $\boldsymbol{A}$ and $\boldsymbol{B}$ at time t_n. Using τ as the weight in a Galerkin weighted residual solution of equation (5.43) leads to a recurrence relationship between a_{n+1} and a_n:

$$\left(\boldsymbol{R}(1-\theta) - \frac{\boldsymbol{S}}{\Delta t}\right)a_n + \left(\boldsymbol{R}\theta + \frac{\boldsymbol{S}}{\Delta t}\right)a_{n+1} + b_n(1-\theta) + b_{n+1}\theta = 0 \qquad (5.47)$$

where $\theta = \frac{2}{3}$.

Adaptive Time Stepping

All drive function start by using a minimum time step. After the first step, the time-step is automatically adjusted to achieve time-stepping relative errors of less than a user supplied tolerance. The time step is increased when the error is much less than the tolerance, or reduced if the error comes close to the tolerance. The time-step can never fall below the minimum time-step. For non-linear problems, a scheme is linked with the solution of equation (5.47).

Fixed Time Step

Alternatively, the user can opt for the fixed time-stepping option, in which case the time-step will remain constant (as defined by the user).

Boundary Conditions

OPERA-2d/TR supports the following types of boundary condition: fixed potential (**Fn=V, Vn=*value***) and periodic boundaries (**Fn=SYMMETRY**). All other boundaries should be set to **Fn=DV, DVn=0** (or to **Fn=NO**, which is equivalent).

Permanent Magnets

Permanent magnets can be specified by a non-zero value of *H* for zero *B* as the first point of a BH curve. The easy direction is given by the region parameter PHASE (see the BHDATA command).

Restarts

OPERA-2d/TR can be restarted from a results file. This allows transient solutions to be continued for larger values of time, or alternatively the Statics (ST) or Steady State AC (AC) analysis programs can provide a solution for restart, enabling 'switch-off' cases to be studied. Statics and AC solutions look like transient solutions at *t*=0.

Preparing an OPERA-2d/TR run

The data required to analyse the model is input using the SOLVE command (see Reference Manual for details). The user can set the following information:

- Linear or Non-linear solution

The linear solution options simply solves with the value of permeability (permittivity) given by the region parameter PERM. A BH curve is still required to give a value for the coercive force. The Non-linear option solves for the non-linear materials as defined by the BH (DE) characteristics defined for material numbers greater than 2. If Non-Linear is chosen the following parameters are set:

  – Number of iterations can be set if a non-linear solution is requested. The value entered sets a limit on the maximum number of non-linear iterations.

  – Tolerance is the convergence tolerance to be applied to the relative change in the solution.

- Time Step - Adaptive or Fixed

  The user has a choice to fix the time step at which each solution is calculated, or set an adaptive time step. In Adaptive Time-stepping integration, a time-step variation is to achieve a user supplied relative tolerance between successive time-steps. In general, values should be between 0.03 and 0.0001, but values outside this range can be given.

- Output Times

  A list of times at which the solution is stored must be supplied.

- Drive function for conductors and circuits

  A drive function for each defined external circuit must be given.

  Any labelled conductor (not in a circuit) can also be given a drive.

  Permanent magnets are DC drive.

  A default drive can be specified for use by any source that has not been explicitly set.

  Non-zero potential boundaries are driven by the region drive (if present), else by the default drive.

  If necessary, the program also prompts for a frequency, phase angle, time constant, time-table file name, etc.

- ‘New solution’ or Restart

  This is only available if the *.op2* file already contains a solution. The TR solution program allows a solution to be continued from the result contained in an existing results file created by one of the analysis programs ST, AC or TR.

Velocity Analysis (VL)

The velocity Analysis Program (OPERA-2d/VL) solves for time invariant magnetic fields including eddy currents induced by motion of conducting media with respect to the field. The model can include non-linear material permeability and either infinite XY or axisymmetric coordinate systems. Linear motion is only allowed in the Y or Z direction. Rotational motion can only take place about the global coordinate system origin. The solution is time invariant, a moving frame of reference is used in the conducting media.

The Equations Solved

OPERA-2d/VL solves for the vector potential defined by a non-linear Poisson equation with a motion induced current term. The equation to be solved is derived by substituting $\frac{1}{\mu}\nabla\times\mathbf{A}$ for **H** in equation (5.18) and including the current induced by uniform motion with respect to the field, this gives

$$\nabla\times\left(\frac{1}{\mu}\nabla\times\mathbf{A}-\mathbf{H}_c\right) = \mathbf{J}+\sigma\mathbf{u}\times\nabla\times\mathbf{A} \qquad (5.48)$$

where **u** is the velocity of the media with respect to the field. The motion induced current term is similar to the advection term in fluid mechanics and a boundary layer effect occurs as the velocity increases.

The formulation of OPERA-2d/VL assumes that, at every instant in time, the geometry of the model is identical. This means that the cross-section of the moving conductor orthogonal to the direction of motion does not change. In other words, for linear motion the moving conductor is “infinite” and for rotational motion the RZ cross-section is invariant. Examples of these might be a pipeline inspection vehicle and an eddy current disk brake respectively.

The solution type must be AT.

Accuracy Guidelines and Upwinding

Meshes where the ‘long’ sides of elements with large aspect ratio are orthogonal to the velocity should be avoided. Also, the use of quadratic elements is not advisable. The solution (without upwinding) will present oscillations if the cell Peclet number *p* is much greater than one. The Peclet number is calculated using

$$p = \mu\sigma\mathbf{u}h \tag{5.49}$$

where h is the size of the element parallel to the velocity, **u**. The oscillations may still occur after mesh refinement. In this case 'upwinding' techniques can be employed to eliminate non-physical 'wiggles' from the solution, although with a possible degradation in accuracy. Such techniques can only be combined with linear elements. Both Hughes method and streamline upwinding are provided, the latter usually being more accurate, but slightly less effective in pathological cases.

OPERA-2d data interpretation

The region parameter CONDUCTIVITY is the conductivity of the media. The region parameter VELOCITY is the Z (axisymmetric) or Y (XY symmetry) velocity or angular velocity (about the global origin) of the media with respect to the field. All the other parameters are treated as in the static field solver. The velocity takes its unit information from the definition of the LENGTH unit.

Length Unit	Velocity Units		
	Axi-symmetry	XY-symmetry	
	Linear Velocity	Linear Velocity	Rotational Velocity
MM	mm s^{-1}	mm s^{-1}	0.1 radian s^{-1}
CM	cm s^{-1}	cm s^{-1}	radian s^{-1}
METRE	m s^{-1}	m s^{-1}	100 radian s^{-1}

Preparing an OPERA-2d/VL run

The data required to analyse the model is input using the SOLVE command (see Reference Manual for details). The user can set the following information:

- Linear or Non-linear solution

  The linear solution options simply solves with the value of permeability (permittivity) given by the region parameter PERM. A BH curve is still required to give a value for the coercive force. The Non-linear option solves for the non-linear materials as defined by the BH (DE) characteristics defined for material numbers greater than 2. If Non-Linear is chosen the following parameters are set:

  – Number of iterations can be set if a non-linear solution is requested. The value entered sets a limit on the maximum number of non-linear iterations.

  – Tolerance is the convergence tolerance to be applied to the relative change in the solution.

- ‘Mesh refinement’ or ‘No mesh refinement’

  This defines whether the mesh is allowed to automatically refine during the analysis. If mesh refinement is selected, the following can be set:

  – Maximum number of iterations

  – Maximum number of elements

  – Final convergence accuracy %

- Scale factor

  The scale factor multiplies charge or current DENSITIES and non-zero boundary condition values. A list of scaling factors can be supplied.

- Upwinding: None, Hughes or Streamline? (N, H or S)

  This prompt allows the user to choose between the upwinding options.The option NONE should be chosen first. If necessary, the results file will contain a recommendation that upwinding should be used, in which case the analysis should be repeated and the results compared carefully.

- Linear or rotational motion

  This parameter allows the user to choose the motion type.

- ‘New solution’ or Restart

  This is only available if the *.op2* file already contains a solution.

Rotating Machines (RM)

The Rotating Machine Program (OPERA-2d/RM) is a Transient Eddy Current Solver, extended to include the effects of rigid body (rotating) motion. The solver also provides for the use of external circuits and coupling to mechanical equations.

OPERA-2d/RM models make use of a ‘Gap Region’ which must be defined by the user to separate the stationary from the rotating part of the model. The Gap Region is a single region, having circular inner and outer radii. Its construction is very simple and is automatically assigned material number 0 (i.e. air with its respective properties). The user defines a value of radius that lies between the inner and outer radii of the Gap Region. The program will create the Gap Region by extending radially inward and outward from the user defined radius until it finds the regions that define the rotor and stator. The faces of the regions that define the inner radius of the stator should be circular, as should those of the outer radius of the rotor. Better results will be obtained from the RM program by including a layer of air elements in the air gap at both the rotor and stator surfaces. The Gap Region then fits between these two layers and ensures that at least 3 elements are used radially across the air gap.

The figures below show the air gap of a simplified 4-pole synchronous machine, showing the separate rotor and stator regions with additional air layers and then the inclusion of the gap region which was specified at a radius of 37.5 cm.

It is possible to exploit the symmetry of the machine. The user specifies a symmetry code when defining the gap region. The symmetry code defines the number of replications of the geometry in 360 degrees, the sign of the symmetry code defines whether the replications exhibit positive or negative periodicity. For example, in the 4-pole machine model shown, a symmetry code of **-4** would be used, since the geometry represents 90 degrees of the complete machine and the poles are alternate North-South. The program automatically applies the correct periodicity boundary conditions for any position to which the rotor has moved.

The Gap Region may also be used advantageously with any other OPERA-2d solver for a rotating electrical machine (or similar) model, since the rotor (or stator) can easily be displaced and the periodicity boundary conditions will be automatically defined.

OPERA-2d/RM models may be constructed with either linear or quadratic elements. However, it should be noted that quadratic elements in the Gap Region only vary linearly in the radial direction. It is advisable to maintain a balanced

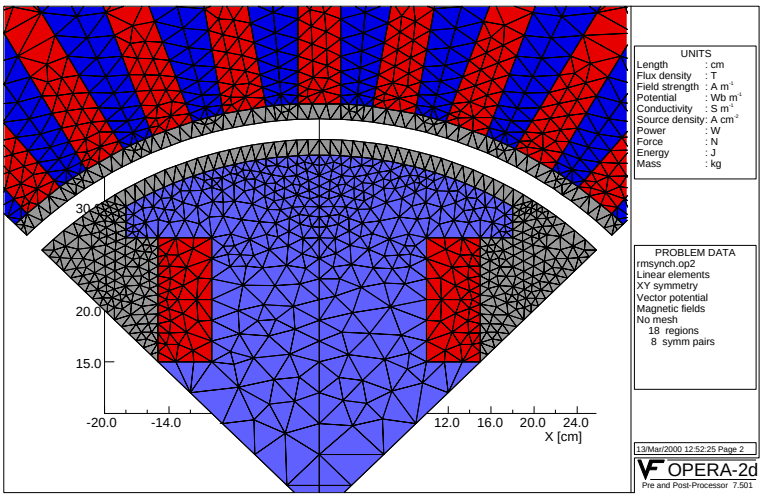


Figure 5.1 Air gap of a simplified 4-pole synchronous machine

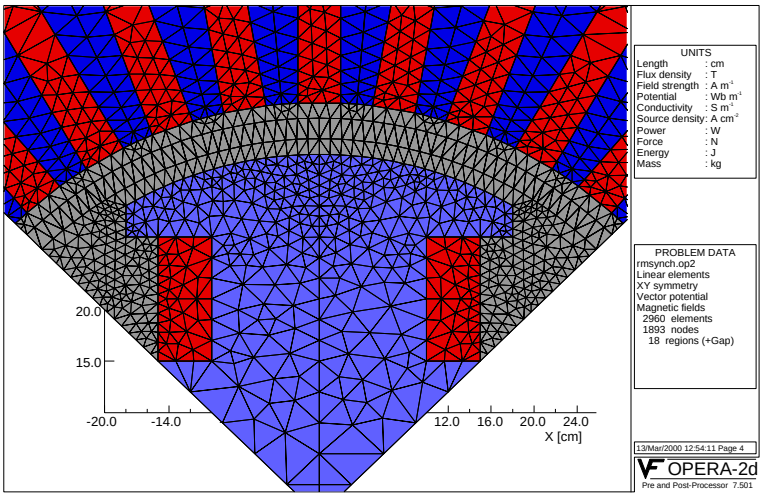


Figure 5.2 Inclusion of the Gap Region

mesh (i.e. mesh of similar number of elements) on either side of the Gap Region (although they do not have to match exactly).

The Equations Solved

OPERA-2d/RM solves the vector diffusion equation with the magnetic vector potential (**`SET SOLUTION=AT`**) as the unknown variable. It is formed from

equations (5.18), (5.21), (5.22) and (5.23), and the integration of (5.19) on page 5-13:

$$\nabla \times \left(\frac{1}{\mu} \nabla \times \mathbf{A} - \mathbf{H}_c \right) = J_s - \sigma \frac{\partial \mathbf{A}}{\partial t} \tag{5.50}$$

in which the current density has been split into the prescribed sources, $\mathbf{J}_s$ and the induced currents, $\sigma \frac{\partial \mathbf{A}}{\partial t}$. In two dimensions, only the z components of $\mathbf{A}$ and $\mathbf{J}_s$ exist. Equation (5.50) can be simplified to:

$$-\nabla \cdot \frac{1}{\mu} \nabla A_z - (\nabla \times \mathbf{H}_c)_z = J_s - \sigma \frac{\partial A_z}{\partial t} \tag{5.51}$$

Driving Functions

The 'driving' field is provided by source currents, J_s, non-zero potential boundary conditions and permanent magnet coercive forces, $\mathbf{H}_c$. The permanent magnets obviously create a dc field. The other driving terms can have a prescribed shape in time. Each value of conductor number (region parameter N) can be related to a different driving function, with, if necessary, every region having a different function.

The driving function options are:

Transient Driving Functions	
Name	**Function**
COSINE	$t<0$: $F = 1$ $t \geq 0$: $F = \cos(2\pi f t - \phi)$. The program prompts for f and ϕ.
DC	Uniform in time from $t=-\infty$ to $t=+\infty$
PEAK	$t<0$: $F = 0$ $t \geq 0 : F = a t \exp\left(\frac{-t^2}{b}\right)$. The program prompts for t_c. a and b are chosen such that $F=1$ at $t=t_c$.

Transient Driving Functions *(continued)*	
RAMP	$t<0$: $F = 0$ $t \geq 0, t \leq t_c$: $F = \frac{t}{t_c}$ $t>t_c$: $F = 1$. The program prompts for t_c.
EXPONENTIAL	$t<0$: $F = 0$ $t \geq 0$: $F = 1 - exp\left(\frac{-t}{t_c}\right)$. The program prompts for t_c.
SINE	$t<0$: $F = 0$ $t \geq 0$: $F = \sin(2\pi ft - \phi)$. The program prompts for f and ϕ.
STEP	$t<0$: $F = 0$ $t \geq 0$: $F = 1$
TTOFF	$t<0$: $F = F(0)$ $t \geq 0$: F = cubic splines The program prompts for the name of a time-table file. F(0) is the value of the function in the table file at time t=0.
TTON	$t<0$: $F = 0$ $t \geq 0$: F = cubic spline The program prompts for the name of a time-table file.

The transient time table option allows the user to define driving functions other than those programmed into the analysis code. The tables consist of files containing up to 1000 pairs of numbers in free format, one pair per line. The first number on each line specifies the time; the second gives the function value. The values of time should start at zero and increase through the file. Discontinuities in function value or first derivative can be forced by specifying two entries for the same value of time. For switch-on cases, the function value at zero time need not be zero, but it is assumed that the function has value zero for all time before zero. Beyond the last value of time in the table the function continues with the same cubic function calculated for the last section of the table.

The driving functions may be viewed using the **Graphs** Option, available from the **FILE** Top Level Menu.

Eddy Current Conductors

In some problems it is necessary to allow eddy currents to flow in driving conductors, redistributing the current density, or to limit the total eddy current in a conductor to zero. This is achieved by solving an extra equation for each group of regions which make up such a conductor. This equation limits the total current, I, flowing in terms of $\frac{\partial A}{\partial t}$ and a potential gradient, ∇V, which comes from the integration of equation (5.19) on page 5-13 and is usually zero.

$$-\int_{\Omega_I} \sigma\left(\frac{\partial A}{\partial t} + \nabla V\right)\partial\Omega = I \tag{5.52}$$

The effect of the potential gradient is a spatially uniform current density over the conductor, J^* given by

$$J^* = -\sigma\nabla V \tag{5.53}$$

and becomes an extra unknown in the modified equation (5.38). The following two equations are solved together

$$-\nabla\cdot\frac{1}{\mu}\nabla A_z - \nabla\times\mathbf{H}_c = J^* - \sigma\frac{\partial A_z}{\partial t} \tag{5.54}$$

$$\int_{\Omega_J}\left(-\sigma\frac{\partial A}{\partial t} + J^*\right)\partial\Omega = \int_{\Omega_J} J_s\partial\Omega \tag{5.55}$$

Equation (5.55) is repeated for every group of regions with a different value of N and SYMMETRY≠0. J_s is given by DENSITY.

External circuits

Transient solutions can be excited either by current sources or by voltage sources connected via external circuits to the model. The current density that can be defined for each region is supplied by a current source, independent of the properties of the coils the current is given as a function of time. The voltage driven option allows a set of coils to be defined as a circuit which is connected to an external voltage source in series with an external resistance, capacitance and inductance. A coil is the set of regions that have the same conductor number N.

The Reference Manual provides a full description of External Circuits options. Hints on the usage of external circuits are given in a separate Application Note.

Time Stepping

The finite element method used in OPERA-2d/RM is similar to that described previously. However the terms in A and $\frac{\partial A}{\partial t}$ in equations (5.38) both yield matrices, referred to as **R** and **S**. The Galerkin procedure leads to a matrix equation

$$\boldsymbol{RA} + \boldsymbol{S}\frac{\partial \boldsymbol{A}}{\partial t} + \boldsymbol{B} = 0 \tag{5.56}$$

where $\boldsymbol{A}$ is now a vector of unknown potentials and $\boldsymbol{B}$ is a vector of driving terms. The solution of equation (5.56) is also based on the Galerkin procedure. $\boldsymbol{A}$ and $\boldsymbol{B}$ are discretized in time using a first order function of time:

$$\boldsymbol{A}(t) = (1-\tau)a_n + \tau a_{n+1} \tag{5.57}$$

$$\boldsymbol{B}(t) = (1-\tau)b_n + \tau b_{n+1} \tag{5.58}$$

where

$$\tau = \frac{t - t_n}{t_{n+1} - t_n} \tag{5.59}$$

and a_n and b_n are values of $\boldsymbol{A}$ and $\boldsymbol{B}$ at time t_n. Using τ as the weight in a Galerkin weighted residual solution of equation (5.56) leads to a recurrence relationship between a_{n+1} and a_n:

$$\left(\boldsymbol{R}(1-\theta) - \frac{\boldsymbol{S}}{\Delta t}\right)a_n + \left(\boldsymbol{R}\theta + \frac{\boldsymbol{S}}{\Delta t}\right)a_{n+1} + b_n(1-\theta) + b_{n+1}\theta = 0 \tag{5.60}$$

where $\theta = 1$.

Adaptive Time Stepping

All drive function start by using a minimum time step. After the first step, the time-step is automatically adjusted to achieve time-stepping relative errors of less than a user supplied tolerance. The time step is increased when the error is much less than the tolerance, or reduced if the error comes close to the tolerance. The time-step can never fall below the minimum time-step. For non-linear problems, a scheme is linked with the solution of equation (5.60).

Fixed Time Step

Alternatively, the user can opt for the fixed time-stepping option, in which case the time-step will remain constant (as defined by the user).

Boundary Conditions

OPERA-2d/RM supports the fixed potential (**Fn=V, Vn=*value***) boundary conditions but does not cater for periodic boundaries. All other boundaries should be set to **Fn=DV, DVn=0** (or to **Fn=NO**, which is equivalent).

Permanent Magnets

Permanent magnets can be specified by a non-zero value of *H* for zero *B* as the first point of a BH curve. The easy direction is given by the region parameter PHASE (see the BHDATA command).

Restarts

OPERA-2d/RM can be restarted from a results file. This allows rotating motion solutions to be continued for larger values of time, or alternatively the Statics (ST), Steady State AC (AC) or Transient (TR) analysis programs can provide a solution for restart, enabling 'switch-off' cases to be studied. Statics and AC solutions look like transient solutions at *t*=0.

Rotor speed

Fixed speed

The simplest assumption in OPERA-2d/RM is that the rotor speed is constant during the duration of the transient analysis. The user specifies the rotation speed in RPM.

Variable speed

The user may specify the speed of the rotor as a function of time during the duration of the transient analysis. For example, the rotational speed (in RPM) of a 4-pole 60 Hz machine rising towards synchronous speed could be given by

$$\text{Speed} = 1800\left(1 - e^{-\frac{t}{0.01}}\right) \tag{5.61}$$

with an exponential rise time constant of 10msec. The user specifies the rotational speed in radians/second using an expression for a CONSTANT called #SPEED in a *.comi* file. To implement equation (5.61), the file contains

```
$CONS #SPEED 1800/60*2*pi*(1-EXP(-TTIME/0.01))
```

where TTIME is the system variable defining the current time in the transient. Clearly, the commands in the file could be considerably more complex with different expressions defining the behaviour at different times by using conditional $IF clauses.

Mechanical coupling

Rather than using a user-specified fixed or varying speed, OPERA-2d/RM can compute the speed based on the rigid body dynamics equation

$$T = J\frac{d^2\theta}{dt^2} \tag{5.62}$$

where T is the torque, J is the polar moment of inertia and θ the angle. The torque is the sum of the electromagnetic torque, computed by the program, and the friction, load and linearly speed-dependent torques, defined by the user.

A second option also exists where the angular acceleration may be defined by the user. This allows conditional behaviour, for instance, where additional load torque may be added to the machine subject to its operating speed or where the load has a time dependent behaviour. Similarly, to the variable speed option, the user defines a CONSTANT called #ACCEL in a *.comi* file. The following system variables that define the mechanical properties are available.

TTIME	Time during transient (seconds)
RMANGLE	Rotational position of rotor (radians)
RMSPEED	Rotational speed (radians/second)
RMTORQUE	Computed electromagnetic torque per unit length
RMLENGTH	Length of the machine
RMINERTIA	Moment of inertia
RMFTORQUE	Frictional torque
RMVTORQUE	Speed dependent torque
RMATORQUE	Applied (load) torque

For example, in the 4-pole machine discussed previously, the user may require that the machine never accelerates above synchronous speed (by applying an additional load torque that exactly cancels the excess electromagnetic torque) i.e. at synchronous speed and higher, the net torque on the rotor is zero. The *.comi* file may contain instructions such as:

```
$IF RMSPEED*60/2/PI GE 1800
 $CONSTANT #TORQUE 0.0
$ELSE
 $CONSTANT #TORQUE RMTORQUE*RMLENGTH-RMATORQUE
$END IF
$CONSTANT #ACCEL #TORQUE/RMINERTIA
```

Output log file

The transient analysis programs store their results at a set of times defined by the user. When the RM solver is run and a long initial transient is expected, it is convenient to be able to observe the progress of the solution. The output from all time steps can be logged to a file in the format required by the GRAPH command. A typical log file would contain the time, rotation angle and speed so that these may be graphed as the solution progresses.

Preparing an OPERA-2d/RM run

The data required to analyse the model is input using the SOLVE command (see Reference Manual for details). The user can set the following information:

- Linear or Non-linear solution

  The linear solution options simply solves with the value of permeability (permittivity) given by the region parameter PERM. A BH curve is still required to give a value for the coercive force. The Non-linear option solves for the non-linear materials as defined by the BH (DE) characteristics defined for material numbers greater than 2. If Non-Linear is chosen the following parameters are set:

  – Number of iterations can be set if a non-linear solution is requested. The value entered sets a limit on the maximum number of non-linear iterations.

  – Tolerance is the convergence tolerance to be applied to the relative change in the solution.

- Time Step - Adaptive or Fixed

  The user has a choice to fix the time step at which each solution is calculated, or set an adaptive time step. In Adaptive Time-stepping integration, a time-step variation is to achieve a user supplied relative tolerance between successive time-steps. In general, values should be between 0.03 and 0.0001, but values outside this range can be given.

- Output Times

  A list of times at which the solution is stored may be supplied.

- Rotation speed - fixed, variable or mechanical coupling

This defines how the rotational speed of the rotating part of the model is determined. (See “Rotor speed” on page 5-37.)

- ‘New solution’ or Restart

  This is only available if the *.op2* file already contains a solution. The TR solution program allows a solution to be continued from the result contained in an existing results file created by one of the analysis programs ST, AC or TR.

  An application note on using the OPERA-2d/RM solver to model an induction motor is included in this manual.

Linear Motion (LM)

The Linear Motion program (OPERA-2d/LM) is a Transient Eddy Current Solver, extended to include the effects of motion. The solution can have XY symmetry, where motion can be in both X and Y directions, as well as allowing for rotational motion about a point. The solution can also have axi-symmetry, where the motion is restricted to motion along the axial Z. The solver also provides for the use of external circuits and coupling to mechanical equations.

OPERA-2d/LM models make use of a special remeshing technique that requires that the user separates the model into 3 groups of regions: the moving regions, the regions that allow motion and the static sections. This process is controlled using the LMMOTION command (see the OPERA-2d/PP reference manual). During the time-stepping solution process the moving section of the model is repositioned and a reconnection mesh between moving and static sections created.

The meshing facilities made available by the LMMOTION command may also be used advantageously with any other OPERA-2d solver for any model, since the part of the model can easily be displaced without requiring as much mesh continuity.

OPERA-2d/LM models may be constructed with either linear or quadratic elements. It is advisable to maintain a balanced mesh (i.e. mesh of similar size elements) in the regions that allow motion and at the boundary of the moving part (although they do not have to match exactly).

The Equations Solved

OPERA-2d/LM solves the vector diffusion equation with the magnetic vector potential (**`SET SOLUTION=AT`**) as the unknown variable. It is formed from equations (5.18), (5.21), (5.22) and (5.23), and the integration of (5.19) on page 5-13:

$$\nabla \times \left(\frac{1}{\mu} \nabla \times \mathbf{A} - \mathbf{H}_c \right) = J_s - \sigma \frac{\partial \mathbf{A}}{\partial t} \qquad (5.63)$$

in which the current density has been split into the prescribed sources, $\mathbf{J}_s$ and the induced currents, $\sigma \frac{\partial \mathbf{A}}{\partial t}$. In two dimensions, only the z components of $\mathbf{A}$ and $\mathbf{J}_s$ exist. Equation (5.63) can be simplified to:

$$-\nabla\cdot\frac{1}{\mu}\nabla A_z-(\nabla\times\mathbf{H}_c)_z = J_s-\sigma\frac{\partial A_z}{\partial t} \qquad (5.64)$$

Driving Functions

The 'driving' field is provided by source currents, J_s, non-zero potential boundary conditions and permanent magnet coercive forces, $\mathbf{H}_c$. The permanent magnets obviously create a dc field. The other driving terms can have a prescribed shape in time. Each value of conductor number (region parameter N) can be related to a different driving function, with, if necessary, every region having a different function.

The driving function options are:

Transient Driving Functions	
Name	**Function**
COSINE	$t<0$: $F=1$ $t\geq 0$: $F=\cos(2\pi ft-\phi)$. The program prompts for f and ϕ.
DC	Uniform in time from $t=-\infty$ to $t=+\infty$
PEAK	$t<0$: $F=0$ $t\geq 0: F = at\,exp\left(\frac{-t^2}{b}\right)$. The program prompts for t_c. a and b are chosen such that $F=1$ at $t=t_c$.
RAMP	$t<0$: $F=0$ $t\geq 0, t\leq t_c: F=\frac{t}{t_c}$ $t>t_c$: $F=1$. The program prompts for t_c.
EXPONENTIAL	$t<0$: $F=0$ $t\geq 0: F = 1-exp\left(\frac{-t}{t_c}\right)$. The program prompts for t_c.
SINE	$t<0$: $F=0$ $t\geq 0$: $F=\sin(2\pi ft-\phi)$. The program prompts for f and ϕ.
STEP	$t<0$: $F=0$ $t\geq 0$: $F=1$

Transient Driving Functions *(continued)*	
TTOFF	t<0: F = F(0) t≥0: F = cubic splines The program prompts for the name of a time-table file. F(0) is the value of the function in the table file at time t=0.
TTON	t<0: F = 0 t≥0: F = cubic spline The program prompts for the name of a time-table file.

The transient time table option allows the user to define driving functions other than those programmed into the analysis code. The tables consist of files containing up to 1000 pairs of numbers in free format, one pair per line. The first number on each line specifies the time; the second gives the function value. The values of time should start at zero and increase through the file. Discontinuities in function value or first derivative can be forced by specifying two entries for the same value of time. For switch-on cases, the function value at zero time need not be zero, but it is assumed that the function has value zero for all time before zero. Beyond the last value of time in the table the function continues with the same cubic function calculated for the last section of the table.

The driving functions may be viewed using the **Graphs** Option, available from the **FILE** Top Level Menu.

Eddy Current Conductors

In some problems it is necessary to allow eddy currents to flow in driving conductors, redistributing the current density, or to limit the total eddy current in a conductor to zero. This is achieved by solving an extra equation for each group of regions which make up such a conductor. This equation limits the total current, I, flowing in terms of $\frac{\partial A}{\partial t}$ and a potential gradient, ∇V, which comes from the integration of equation (5.19) on page 5-13 and is usually zero.

$$-\int_{\Omega_I} \sigma\left(\frac{\partial A}{\partial t} + \nabla V\right)\partial\Omega = I \qquad (5.65)$$

The effect of the potential gradient is a spatially uniform current density over the conductor, J^* given by

$$J^* = -\sigma\nabla V \qquad (5.66)$$

and becomes an extra unknown in the modified equation (5.64). The following two equations are solved together

$$-\nabla\cdot\frac{1}{\mu}\nabla A_z - \nabla\times\mathbf{H}_c = J^* - \sigma\frac{\partial A_z}{\partial t} \tag{5.67}$$

$$\int_{\Omega_J}\left(-\sigma\frac{\partial A}{\partial t} + J^*\right)\partial\Omega = \int_{\Omega_J} J_s\partial\Omega \tag{5.68}$$

Equation (5.68) is repeated for every group of regions with a different value of N and SYMMETRY≠0. J_s is given by DENSITY.

External circuits

Transient solutions can be excited either by current sources or by voltage sources connected via external circuits to the model. The current density that can be defined for each region is supplied by a current source. Independent of the properties of the coils the current is given as a function of time. The voltage driven option allows a set of coils to be defined as a circuit which is connected to an external voltage source in series with an external resistance, capacitance and inductance. A coil is a set of regions that have the same conductor number N.

The Reference Manual provides a full description of External Circuits options. Hints on the usage of external circuits are given in a separate Application Note.

Time Stepping

The finite element method used in OPERA-2d/LM is similar to that described previously. However the terms in A and $\frac{\partial A}{\partial t}$ in equations (5.64) both yield matrices, referred to as **R** and **S**. The Galerkin procedure leads to a matrix equation

$$\boldsymbol{RA} + \boldsymbol{S}\frac{\partial \boldsymbol{A}}{\partial t} + \boldsymbol{B} = 0 \tag{5.69}$$

where $\boldsymbol{A}$ is now a vector of unknown potentials and $\boldsymbol{B}$ is a vector of driving terms. The solution of equation (5.69) is also based on the Galerkin procedure. $\boldsymbol{A}$ and $\boldsymbol{B}$ are discretized in time using a first order function of time:

$$\boldsymbol{A}(t) = (1-\tau)a_n + \tau a_{n+1} \tag{5.70}$$

$$\boldsymbol{B}(t) = (1-\tau)b_n + \tau b_{n+1} \tag{5.71}$$

where

$$\tau = \frac{t - t_n}{t_{n+1} - t_n} \tag{5.72}$$

and a_n and b_n are values of $\boldsymbol{A}$ and $\boldsymbol{B}$ at time t_n. Using τ as the weight in a Galerkin weighted residual solution of equation (5.69) leads to a recurrence relationship between a_{n+1} and a_n:

$$\left(\boldsymbol{R}(1-\theta) - \frac{\boldsymbol{S}}{\Delta t}\right)a_n + \left(\boldsymbol{R}\theta + \frac{\boldsymbol{S}}{\Delta t}\right)a_{n+1} + b_n(1-\theta) + b_{n+1}\theta = 0 \tag{5.73}$$

where $\theta = 1$.

Adaptive Time Stepping

All drive functions start by using a minimum time step. After the first step, the time-step is automatically adjusted to achieve time-stepping relative errors of less than a user supplied tolerance. The time step is increased when the error is much less than the tolerance, or reduced if the error comes close to the tolerance. The time-step can never fall below the minimum time-step. For non-linear problems, a scheme is linked with the solution of equation (5.73).

Fixed Time Step

Alternatively, the user can opt for the fixed time-stepping option, in which case the time-step will remain constant (as defined by the user).

Boundary Conditions

OPERA-2d/LM supports the fixed potential (**`Fn=V, Vn=value`**) boundary conditions but does not cater for periodic boundaries. All other boundaries should be set to **`Fn=DV, DVn=0`** (or to **`Fn=NO`**, which is equivalent).

Permanent Magnets

Permanent magnets can be specified by a non-zero value of *H* for zero *B* as the first point of a BH curve. The easy direction is given by the region parameter PHASE (see the BHDATA command).

Restarts

OPERA-2d/LM can be restarted from a results file. This allows linear motion solutions to be continued for larger values of time, or alternatively the Statics (ST), Steady State AC (AC) or Transient (TR) analysis programs can provide a solution for restart, enabling 'switch-off' cases to be studied. Statics and AC solutions look like transient solutions at t=0.

Motion

Variable speed

The user may specify the speed of the moving part as a function of time during the duration of the transient analysis. For example, the rotational speed (in RPM) of a 4-pole 60 Hz machine rising towards synchronous speed could be given by

$$\text{Speed} = 1800\left(1 - e^{-\frac{t}{0.01}}\right) \tag{5.74}$$

with an exponential rise time constant of 10msec. The user specifies the rotational speed in radians/second using an expression for a CONSTANT called #ROTSPEED in a *.comi* file. To implement equation (5.74), the file contains

```
$CONS #ROTSPEED 1800/60*2*pi*(1-EXP(-TTIME/0.01))
$CONS #SPEEDX 0.0
$CONS #SPEEDY 0.0
```

where TTIME is the system variable defining the current time in the transient. Clearly, the commands in the file could be considerably more complex with different expressions defining the behaviour at different times by using conditional $IF clauses and output values from the solver (a list of these is given in the table in "Output log file" on page 5-47).

The following is a table defining the list of variables that should be defined within the file for controlling the speed.

XY symmetry, variable velocity	
#SPEEDX	Speed in the X direction (length units / s)
#SPEEDY	Speed in the Y direction (length units / s)
#ROTSPEED	Rotational speed (rad/s)

Axi-symmetry, variable velocity	
#SPEEDZ	Speed in the Z direction (length units / s)

Mechanical coupling

Rather than using a user-specified fixed or varying speed, OPERA-2d/LM can compute the speed based on values of acceleration. Both the linear and angular acceleration may be defined by the user. This allows conditional behaviour, for instance, where additional load force / torque may be added to the machine subject to its operating speed or where the load has a time dependent behaviour. Similarly, to the variable speed option, the user defines CONSTANTS that control the acceleration at each time.

XY symmetry, mechanical coupling	
#ACCELX	Acceleration in the X direction (length units / s^2)
#ACCELY	Acceleration in the Y direction (length units / s^2)
#ROTACCEL	Rotational acceleration (rad/s^2)

Axi-symmetry, mechanical coupling	
#ACCELZ	Acceleration in the Z direction (length units / s^2)

For example, a body being accelerated in the X direction could be controlled using

```
$CONSTANT #ROTACCEL 0.0
$CONSTANT #ACCELY 0.0
$CONSTANT #MASS 10.0
$CONSTANT #ACCELX LMXFORCE/#MASS
```

In this example, the only force acting on the moving body is the electromagnetic force `LMXFORCE`.

Output log file

The transient analysis programs store their results at a set of times defined by the user. When the LM solver is run and a long initial transient is expected, it is convenient to be able to observe the progress of the solution. The output from all time steps can be logged to a file in the format required by the GRAPH command. A typical log file would contain the time, speed and forces so that these may be graphed as the solution progresses.

A set of these outputs is defined when setting the analysis options. The outputs can be any of the following values (or expressions based upon them)

General variables	
TTIME	Current solution time
I1, I2, ... , IN	Current flowing in each external circuit defined in the model.

Variables for XY symmetry	
LMXSHIFT	Displacement in the X direction (length units)
LMYSHIFT	Displacement in the Y direction (length units)
LMROTANGLE	Rotational displacement (degrees)
LMXSPEED	Speed in the X direction (length units / s)
LMYSPEED	Speed in the Y direction (length units / s)
LMROTSPEED	Rotational speed (rad/s)
LMXFORCE	Force in the X direction (Force unit / length unit)
LMXFORCE	Force in the Y direction (Force unit / length unit)
LMTORQUE	Torque on the moving part (Force units)

Variables for axi-symmetry	
LMZSHIFT	Displacement in the Z direction (length units)
LMZSPEED	Speed in the Z direction (length units / s)
LMZFORCE	Force in the Z direction (Force units)

Preparing an OPERA-2d/LM Model

Arranging Groups

The preparation of the finite element model for the LM Solver requires that the model is separated into 3 groups of regions, namely the MOVINGGROUP, the MEDIUMGROUP and the STATIC regions. The names for the former two groups are chosen by the user and subsequently assigned their special properties using the LMMOTION command.

Figure 5.3 shows a primitive linear motor. Three phase windings, arranged in their respective slots, are excited sequentially to attract the moving part into place. Symmetry is exploited in this model, hence the modelling of half the machine.

The moving part region together with thin slices of air surrounding it are grouped under the name MOVE. The air regions along which the moving part will slide are grouped under the name ENV. Figure 5.4 and Figure 5.5 illustrate the members of each Group.

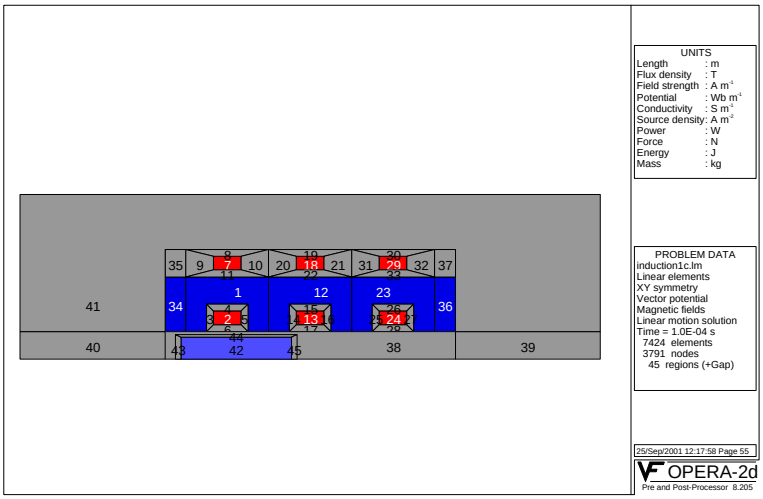


Figure 5.3 A simple linear motor model

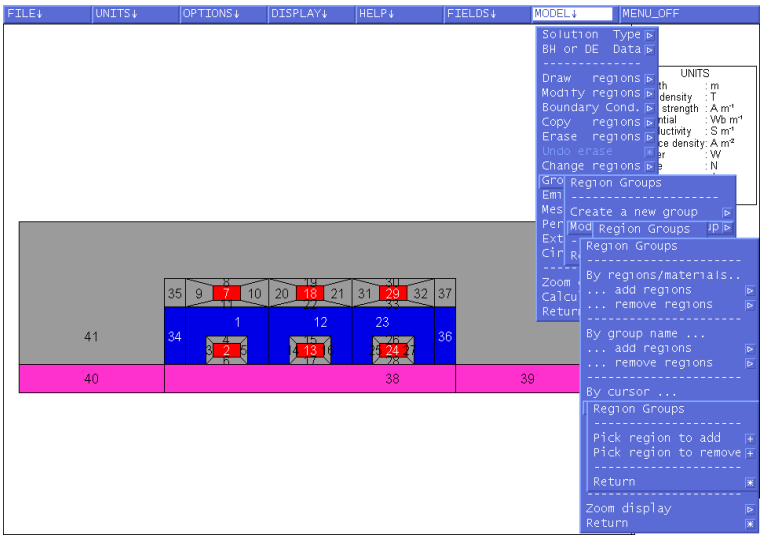


Figure 5.4 Members of the `MEDIUMGROUP`

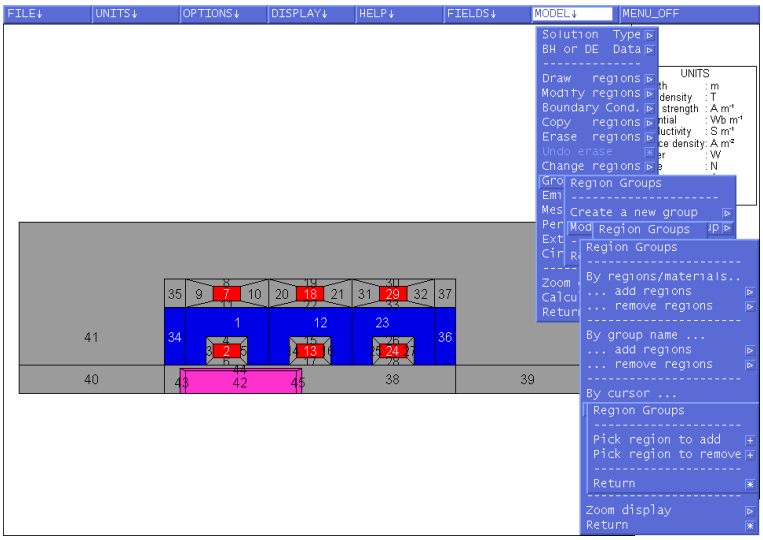


Figure 5.5 Members of the `MOVINGGROUP`

Modelling aspects of the `MOVINGGROUP`

The regions defining the **`MOVINGGROUP`** are drawn on top of the regions defining the **`MEDIUMGROUP`**. This unusual operation can only be valid with the scope of subsequently using the LMMOTION command. If symmetry is exploited in the model, such as the case of the present example, drawing regions of the **`MOVINGGROUP`** will result in the user being prompted to add points to the underlying regions. The user must reply **`NO`** to this.

In its simplest form, the MOVE Group would only contain one region, namely the moving part. However, in order to obtain accurate answers on force calculations, there should be at least three layers of elements in the air gap between the stator and moving part. The latter is therefore surrounded by air regions with Q-shaped elements which are added to the MOVE Group, ensuring a finer discretisation in the region of interest. Figure 5.6 shows the mesh around the plunger area.

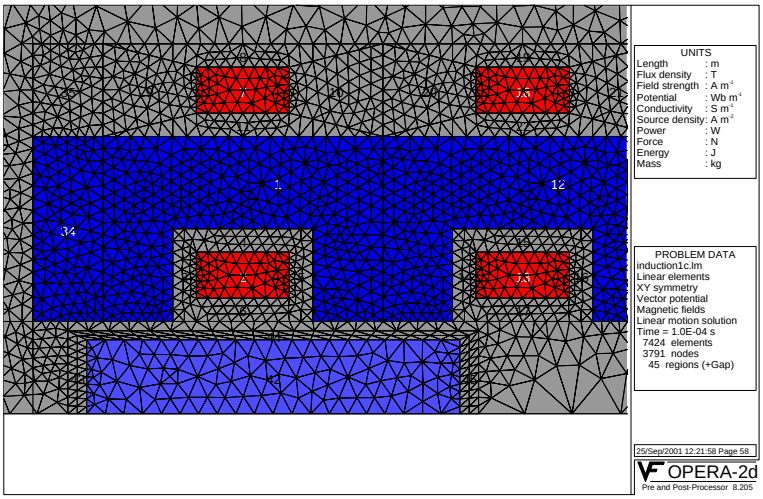


Figure 5.6 Mesh around the moving region

The LMMOTION command

The MOVE and ENV regions are assigned their special properties by selecting

`Model -> Draw Regions -> Linear motion air gap`

and completing the dialog box shown in Figure 5.7.

In keyboard mode the command is

```
LMMOTION MESHING=YES MOVINGGROUP=MOVE MEDIUMGROUP=ENV
CENTREX=0 CENTREY=0
```

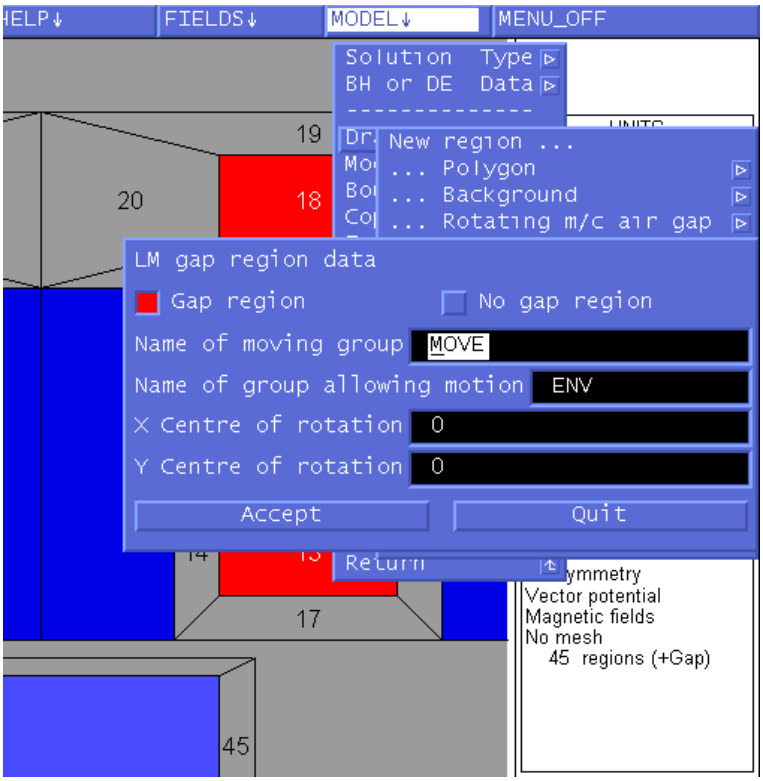


Figure 5.7 Assigning special properties

Variable speed or Variable acceleration

The LM solver offers a variety of operation modes through the use of command (**.comi*) files. Command files can be used to assign a variable velocity; this is applicable in cases where a function of the velocity is known. Here is an example:

```
$CONS #SPEEDX 100*COS(2*PI*1000*TTIME)
$CONS #SPEEDY 0
$CONS #ROTSPEED 0
```

This will actually result in the moving part performing an oscillatory motion along the +/- X direction.

Alternatively, a comi file can also be used to assign a variable acceleration, based on the force exerted on the moving part at any given time. An example of this follows:

```
$CONS #ACCELY 0
$CONS #ROTACCEL 0
$CONS #MASS 0.1
```

```
$CONS #SPRINGFORCE LMXSHIFT*5.2E-03
$CONS #ACCELX (LMXFORCE-#SPRINGFORCE)/#MASS
```

In this example, the total accelerating force acting on the moving part is a function of the electromagnetic force exerted on it, as well as the mechanical force acted upon it by the spring to which it is attached.

Preparing an OPERA-2d/LM run

The data required to analyse the model is input using the SOLVE command (see Reference Manual for details). The user can set the following information:

- Linear or Non-linear solution

  The linear solution options simply solves with the value of permeability (permittivity) given by the region parameter PERM. A BH curve is still required to give a value for the coercive force. The Non-linear option solves for the non-linear materials as defined by the BH (DE) characteristics defined for material numbers greater than 2. If Non-Linear is chosen the following parameters are set:

  – Number of iterations can be set if a non-linear solution is requested. The value entered sets a limit on the maximum number of non-linear iterations.

  – Tolerance is the convergence tolerance to be applied to the relative change in the solution.

- Time Step - Adaptive or Fixed

  The user has a choice to fix the time step at which each solution is calculated, or set an adaptive time step. In Adaptive Time-stepping integration, a time-step variation is to achieve a user supplied relative tolerance between successive time-steps. In general, values should be between 0.03 and 0.0001, but values outside this range can be given.

- Output Times

  A list of times at which the solution is stored may be supplied.

- Motion - variable speed or mechanical coupling

  This defines how the motion of the moving part of the model is determined. (See “Motion” on page 5-46.)

- ‘New solution’ or Restart

  This is only available if the *.op2* file already contains a solution. The TR solution program allows a solution to be continued from the result contained in an existing results file created by one of the analysis programs ST, AC or TR.

Space Charge Beam Analysis (SP)

The Space Charge Beam Analysis Program (OPERA-2d/SP) solves for time invariant electric fields including the effects of space charge created by beams of charged particles. Infinite XY or axisymmetric coordinate systems can be used. A magnetic field may also be applied in addition to the electric field.

The Equations Solved

OPERA-2d/SP solves for the electric scalar potential defined by Poisson's equation. The scalar potential form is derived by substituting $\varepsilon\nabla\phi$ in the equation defining the electric flux density ($\nabla\mathbf{D} = \rho$) to give

$$\nabla \cdot \varepsilon(\nabla\phi - \mathbf{E}_c) = \rho \qquad (5.75)$$

Where ρ is the electric charge density. The term involving $\mathbf{E}_c$ represents electrets.

OPERA-2d offers a number of choices of solution potential that can be selected by using the SET command. In order to use the space charge analysis program the model must be created with the following options

```
SET SOLUTION=V FIELD=ELECTRIC
```

The space charge density can be specified as a region property; however, in the main application of this analysis program, the space charge is associated with beams of charged particles. To calculate the space charge and the resulting electric field the following procedure is followed:

1. The charged particle emitting surfaces are specified by the user.
2. The analysis program computes the trajectories of the emitted particles in the electric (and magnetic) fields. Each particle trajectory is attached to an area on the emitter surface and a current can therefore be associated with it. Note that the particle trajectory calculations include full relativistic correction.
3. The space charge in the beams is included in the electric field recalculation.
4. This iteration (steps 2 and 3) is repeated until the electric field converges to the correct solution.

Different types of particle emission models can be selected in the analysis program. These include Child's law and Langmuir/Fry relationships for the calculation of the space charge limited current, field effect emission and defined current densities and initial energy. Several emitting surfaces can be specified in a model,

the type of particle and the emission model used can be selected for each surface. The emitting surfaces may overlap each other.

OPERA-2d Data interpretation

The OPERA-2d/SP analysis program expects material code number 0 to be free space with zero charge density, material number 1 to have charge density not equal to zero and relative permittivity of one. The region values of PERM are used and the charge density parameters may be non zero. DE (BH) characteristics are required because the value of **E** for the first point of the table (which must have **D** equal zero) defines the $\mathbf{E}_c$ for the material.

The region parameter PHASE defines the easy direction of polarization for an electret material. The remanent polarization is in the negative X direction of a local coordinate system that is rotated anticlockwise by an angle PHASE in degrees from the X axis.

In general, the calculated trajectories of particles are stopped at the boundary of the finite element mesh, but not at the axis of axisymmetric models. Stopping the trajectory calculation in this way is computationally expensive and, in some applications there may be a requirement for the particles to be stopped interior to the mesh. Trajectory calculations are therefore stopped when the trajectory intersects a region with material code 1. It is recommended that regions with material code 1 are added to the outside of the mesh where the beam leaves the model in order to improve the efficiency of the calculation.

Thermal emission models

Thermal saturation limit - type 0

The thermal saturation limit assumes that the electron current emitted by an electrode is independent of the applied voltage, and that it depends only on the temperature, work function and emission constant of the material. The current density is given by the Richardson-Dushman law

$$j_e = \mathrm{AT}^2 e^{\frac{-q_e \phi_w}{\mathrm{kT}}} \tag{5.76}$$

where A is the emission constant for the surface in Amps/cm^2, ϕ_w is the work function of the cathode material in volts, q_e is the electronic charge in coulombs, k is Boltzmann's constant (1.3804×10^{-23} Joule/Kelvin) and T is the temperature of the cathode.

The initial velocity of the particles is assumed to be the mean of the thermal velocity Maxwellian distribution.

Child's Law current limit - type 2

Child's law gives the maximum current density that can be carried in a beam of charged particles across a one dimensional accelerating gap. The equation is derived by requiring equilibrium of the charged particles with a self consistent space charge field.

In order to apply this equation within the program, an accelerating gap width d must be supplied, see Figure 5.8. The equation only applies to infinite planar emitters. It is assumed that the radius of curvature of the emitting surface is large compared to the dimension *d* and that therefore the one dimensional solution can be used.

The equation for the space charge limited current is

$$j_e = \frac{4\varepsilon_0}{9}\sqrt{\frac{2Zq}{m_0}}\frac{V_0^{3/2}}{d^2} \tag{5.77}$$

where ε_0 is the permittivity of free space, Zq is the charge on the particle in coulombs, m_0 is the particle rest mass in kilograms and V_0 is the accelerating voltage applied to the accelerating gap d.

When this model of the emitter is used in the space charge solution program, the accelerating voltage is determined by calculating the voltage at a distance *d* normal to emitter surface, from the finite element solution.

The distance *d* should be such that two or three elements are included between the sample point and the surface, it should also be small compared to the radius of curvature of the surface.

The initial velocity of the particles is assumed to be the mean of the thermal velocity maxwellian distribution. The emission constant, work function and temperature must be given, and if the Child's law current density exceeds the thermal saturation limit, the current density will be limited to the saturation value.

For emitter types 1, 2 and 8 the current from the emitter is equal to the current actually leaving the emitter. Any test beamlet (particle) that is returned to the emitter by the potential barrier is not included in the total current from the emitter that the program prints out in the *.res* file.

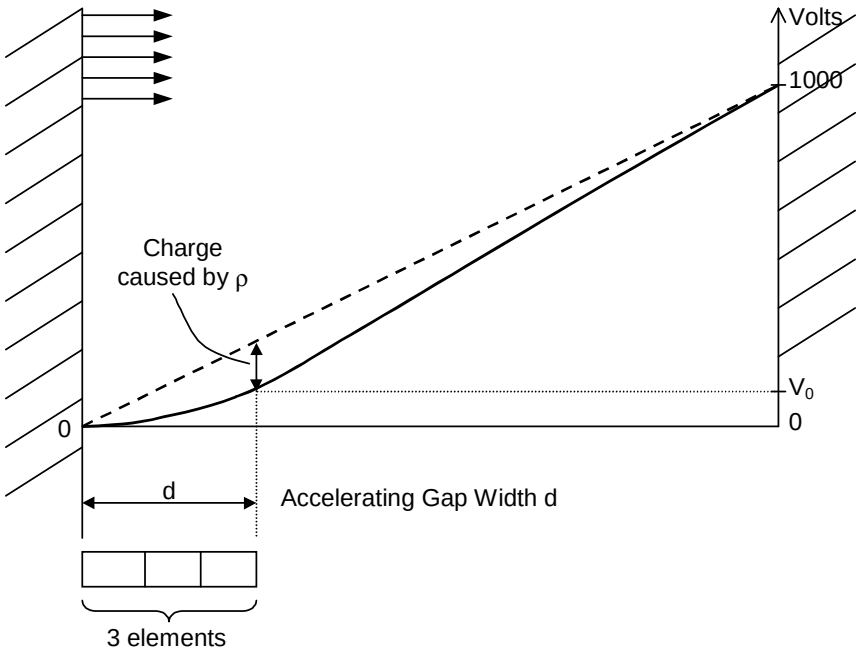


Figure 5.8 Potential distribution between two electrodes for a Child's Law emitter. The emitter on the left-hand side has a potential of 0 volt. The other electrode on the right-hand side has a potential of 1000 volt.

Langmuir/Fry current limit - types 1 and 8

Child's law assumes zero initial energy particles. A more realistic one dimensional solution can be found taking into account the velocity distribution of particles in a thermionic emitter.

In many cases thermionic emitters are operated in a space charge limited mode. This produces a uniform current distribution because the flow is insensitive to local variations of the surface emissivity. The initial particle energy distribution is important in this case.

A non-linear differential equation must be solved in order to calculate the space charge limited current (Reference: Kirstein, Kino & Waters, Space Charge Flow, McGraw-Hill, pp265-276), the program solves the non-linear equation using a shooting method.

As in the Child's law model, a sample distance *d* must be specified, see Figure 5.9. The sampling distance should be 2 to 3 times the value of the potential minimum distance.

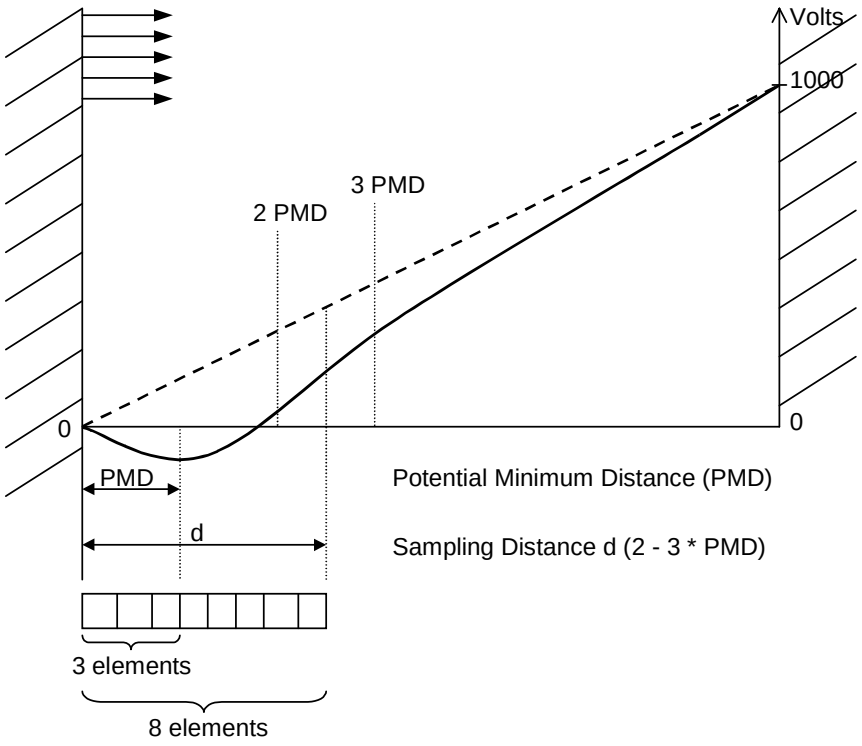


Figure 5.9 Potential distribution between two electrodes. The emitter on the left-hand side has a potential of 0 volt. The other electrode on the right-hand side has a potential of 1000 volt. The space charge in front of the emitter is generating a potential barrier.

The one dimensional Langmuir/Fry model is then solved using the voltage computed at the sample point.

The initial velocity of the particles is computed from the displaced Maxwellian distribution of the particles that escape the voltage minimum that exists in front of the emitter.

Two options are supported with the Langmuir/Fry emission model. The particles are either tracked from the specified emission surface (type 8) or from a virtual cathode (type 1).

The virtual cathode is defined by a set of positions on the normals from points on the emission surface, where the voltage is equal to the value at the emission surface. This assumes that the space charge is creating a voltage minimum in front of the emitter, if the voltage minimum does not exist then particles are tracked from the emitter surface.

It should be noted that the sample distance *d* must be larger than the spacing between the emission surface and the computed virtual cathode.

For emitter types 1, 2 and 8 the current from the emitter is equal to the current actually leaving the emitter. Any test beamlet (particle) that is returned to the emitter by the potential barrier is not included in the total current from the emitter that the program prints out in the *.res* file.

Maxwell velocity distribution sampling – type 10 and 11

This option samples the velocity distribution of the electrons escaping from a thermionic emitter. Normal and tangential velocities may be sampled in the current release of the software.

The Maxwellian velocity distribution is divided into a set of ranges (also called bins) such that each range contains the same number of particles. Figure 5.10 shows an example of a velocity distribution with 4 bins. A test ray is tracked from each velocity bin, its velocity is determined by the average kinetic energy of the particles in the bin.

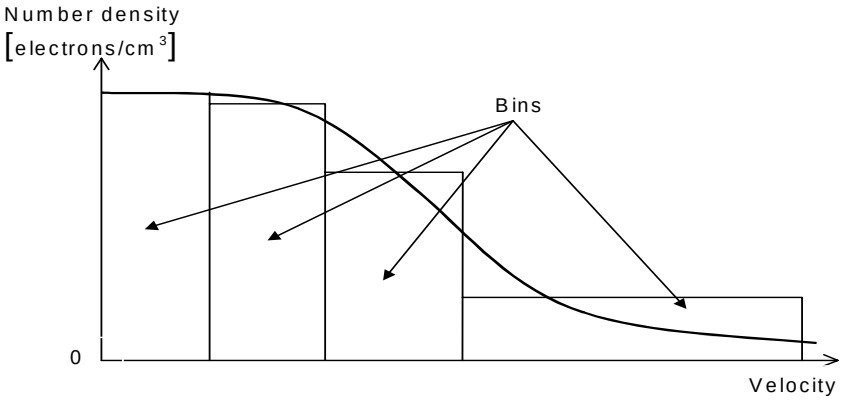


Figure 5.10 Maxwell velocity distribution with 4 ranges

Emitter type 10 uses only a normal velocity sampling; emitter type 11 uses both normal and tangential velocity sampling (the Maxwell distributions for the normal and tangential directions are independent).

The current density of electrons at a particular velocity (v) is $j_{e*}\, n(v)$ where

$$j_e = \mathrm{AT}^2\, e^{\frac{-q_e \phi_w}{\mathrm{kT}}} \tag{5.78}$$

and the normalised distribution function in terms of velocity is

$$n(v) = 2\sqrt{\frac{m}{2\pi kT}}\, e^{\frac{-mv^2}{2kT}} \tag{5.79}$$

Depending on the specific geometry of a finite element model some space charge may build up in front of an emitter. In this case the cathode's current is “space charge limited”; there will be a potential barrier in front of the cathode created by the space charge, see Figure 5.11.

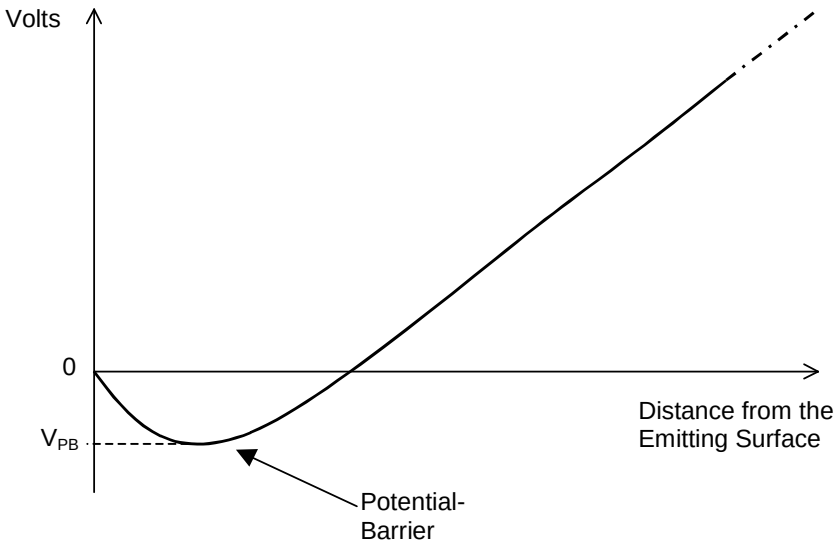


Figure 5.11 Potential barrier in front of an emitter

In order to leave the emitter surface, an electron needs an initial velocity (kinetic energy) greater than that required to pass over the potential barrier.

If the current is space charge limited, the finite element mesh close to the emitter surface must be capable of modelling the voltage minimum that may occur.

With a given value of the potential barrier V_{PB} the escape-velocity can be worked out, see Figure 5.12. In the given example four test trajectories are being emitted per emitting point, but only two of these are capable of passing the potential barrier.

A test ray from a discrete normal velocity bin can escape the potential barrier only if

$$v_{normal} > \frac{2 \cdot q \cdot V_{PB}}{m_e} \tag{5.80}$$

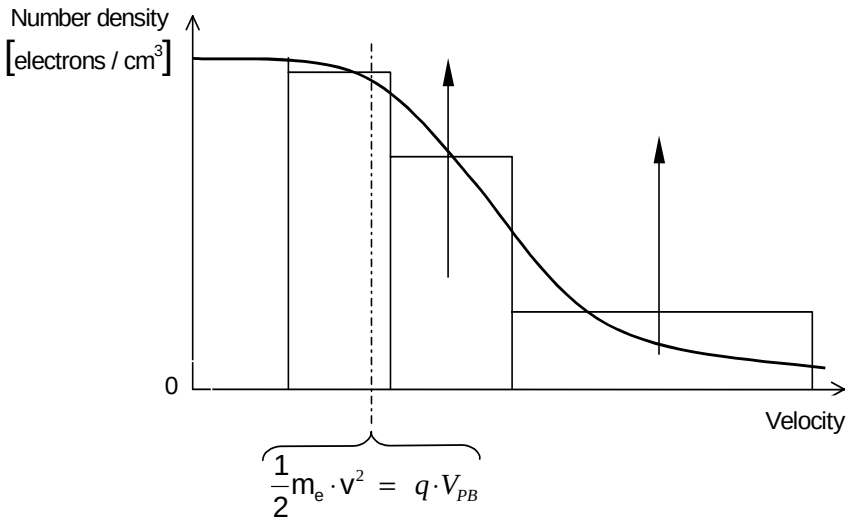


Figure 5.12 Two trajectories with energy greater than V_{PB} are leaving the example-emitter of Figure 5.10

The potential barrier doesn't affect the tangential velocities of emitter type 11. The only fact which determines whether a particle escapes or not is the normal velocity.

Example for emitter type 11

If you ask the program for a normal sampling = 4 and a tangential sampling = 2, the Maxwell velocity distribution will give 4*2*2*4 = 64 test-rays per emitting point, but only 32 of these can escape the potential barrier in Figure 5.12.

The factors above are based on

- 4 normal samplings for the example in Figure 5.10,
- 2 beams tangential (in x direction),
- 2 beams tangential (in y direction) and
- 4 combinations (x+y, x-y, -x+y, -x-y).

The resolution of the emission current is defined by the equation

$$Resolution = \frac{I_{Saturation}}{number\ of\ v_{normal}\ bins} \tag{5.81}$$

For a full nonlinear solution a sampling of the normal velocity should be expected to require more than 10 bins.

Field effect emission models

Fowler Nordheim Field emission - type 4

This option calculates the current that will be extracted by very high electric fields applied to the surface of the cathode. Significant tunnelling of electrons close to the surface of a cold metallic cathode will occur if a high electric field is applied to materials with a low thermionic work function. Fowler and Nordheim (1928) derived a rigorous solution for this tunnelling current:

$$J_e = 6.2 \times 10^2 \left(\frac{E_f}{\phi_w}\right)^{\frac{1}{2}} \frac{E^2}{E_f + \phi_w} \, e^{\frac{-6.83 \times 10^7 \phi_w^{\frac{3}{2}}}{E}} \tag{5.82}$$

where J_e is the current density in *Amps/cm*^{2}, E_f is the Fermi energy of electrons in the metal and E is the electric field in *Volts/Metre* applied to the metallic surface. Extensions to the basic theory to include the change in barrier voltage produced by the escaping electrons give the following result:

$$J_e = A\left(\frac{hqE}{4\pi K}\right)^2 \cdot \frac{1}{2qm_0\phi_w} \, e^{\frac{4\pi b(2qm_0)^{1/2}\phi_w^{3/2}}{Eh}} \tag{5.83}$$

where h is Planck's constant, and:

$$b = \frac{2\left(1 - \left(\frac{(qE)^{\frac{1}{2}}}{(4\pi\varepsilon_0)^{\frac{1}{2}}\phi_w}\right)^{1.69}\right)}{3\left(1 + 0.1107\left(\frac{(qE)^{\frac{1}{2}}}{(4\pi\varepsilon_0)^{\frac{1}{2}}\phi_w}\right)^{1.33}\right)} \tag{5.84}$$

Schottky Field emission - type 5

This option calculates the current that will be emitted from a cathode, at a known temperature, for the lowest field strengths at which field emission occurs. The current density (J_s in *Amps/cm*^{2}) is given by

$$J_s = AT^2 e^{\frac{-q_e\phi_w + (q_e^3 E)^{\frac{1}{2}}}{KT}} \quad (5.85)$$

Extended Schottky Field emission - type 6

At intermediate fields, the extended Schottky model predicts higher and more realistic currents. In this case the current density is given by

$$J_e = J_s \frac{\frac{\pi h q_e^{1/4} E^{3/4}}{\pi (2m)^{1/2} KT}}{\sin \frac{\pi h q_e^{1/4} E^{3/4}}{\pi (2m)^{1/2} KT}} \quad (5.86)$$

where J_s is given by equation 5.85

Automatic Field Emission selection - type 7

Three regimes of field emission are available in the software, the choice depends on the magnitude of the electric field at the start point of each particle. Users may either select a particular model, as shown above, or allow the program to determine which model is appropriate (type 7). The program will select the model that gives the largest current density.

Other emission models

Specified current density - type 3

The program has options that allows a current density and initial particle energy to be specified.

Plasma Free Surface Model - type 102

A simple model has been introduced that allows extraction of particles from plasmas to be modelled. The emission boundary of the plasma is defined as the surface on which the normal component of the electric field strength is constant. The first point of the user specified emitter surface is kept fixed, the other points are adjusted to achieve this constant field condition. Child's law is then used to compute the current density that can be extracted from each point on the plasma emission boundary.

Emitter Data File

The emitter surfaces must be described in a data file with a file name extension of *emit* and the same file name as the model geometry file. This file can be created using the pre and post processor EMIT command or can be created using a text editor with the format described below (N.B. CGS units must be used in the emitter data file). The emitter datafile format is as follows (note that free format input is used and all data items must therefore be entered).

Record 1 Model symmetry

This data is required to provide compatibility with the 3D version of the space charge beam software. The data is not at present used in the 2D program.

Field Number	Type	Units	Description
1	Integer	None	Order of rotational symmetry about the Z axis. (Not yet used in OPERA-2d.)
2	Integer	None	Reflection flag for XY plane. (Not yet used in OPERA-2d.)
3	Integer	None	Reflection flag for YZ plane. (Not yet used in OPERA-2d.)
4	Integer	None	Reflection flag for ZX plane. (Not yet used in OPERA-2d.)

Record 2 Global parameters

A number of independent emitting surfaces may be defined in a model. Each surface has its own emission characteristics. Some parameters apply to all aspects of the calculation, for example, the maximum distance between the points used to represent each trajectory, the accuracy of the trajectory calculation and the distance used to evaluate the current limit models.

Field Number	Type	Units	Description
1	Integer	None	Number Of Emitters
2	Real	cm	Maximum step length allowed in the trajectory calculation.

3	Real	cm	Absolute tolerance for the trajectory calculation.
4	Real	cm	Normal sampling distance. The distance from the emitter surface used to sample the potential for the Child's and Langmuir/Fry equations.

Records 3 to 8 are then repeated as a group, NUMBER of EMITTER times.

Record 3 Emitter characteristics

The emission model is specified for each emitter, together with the data required to characterize the emitter.

Field Number	Type	Units	Description
1	Integer	None	Emitter type:
			0 = Thermal saturation limit
			1 = 1D Langmuir/Fry limit (with virtual cathode)
			2 = 1D Child's law limit
			3 = specified current density
			4 = Fowler Nordheim Field emission
			5 = Schottky Field emission
			6 = Extended Schottky field emission
			7 = Automatic selection between types 4, 5 and 6
			8 = 1D Langmuir/Fry limit
			10= Maxwell Normal Velocity sampling
			11= Maxwell Normal and Tangential Velocity sampling
			102= Plasma free surface model
2	Real	kelvin	Emitter temperature
3	Real	volt	Emitter Work function (or initial particle energy, emitter type 3)
4	Real	amp cm^{-2}	Emission constant for the emitter (for types 0, 1, 2, 5, 6 and 10).
			Current density for type 3 emitter.

Record 4 Particle type

Field Number	Type	Units	Description
1	Real	None	Particle rest mass in electron rest mass units.
2	Integer	None	Number of charge quanta carried by the particle (-1 for an electron).

Record 5 Sample rays

Enhanced models for the emitters are being developed, these will include sampling of the velocity distribution of the input particles. Trajectories are calculated for a set of particles. In the curved line segment emitter models at least one particle is started from each subdivision of the line. The maximum distance parameter can be used to increase the number used.

Field Number	Type	Units	Description
1	Integer	None	Number of sample bins used to sample tangential velocity (only used with emitter type 11).
2	Integer	None	Number of sample bins used to sample normal velocity (only used with emitter types 10 and 11).
3	Real	cm	Maximum distance (tangential to the emitter surface) between sample rays.

Record 6 Number of line segments representing the emitter

Field Number	Type	Units	Description
1	Integer	None	The number of line segments in the model for the emitter.

Records 7 and 8 are repeated as a group, number of line segments times.

Record 7 **Line segment geometry type**

Field Number	Type	Units	Description
1	Integer	None	Geometry type for the line segment (0 = curved line segment).

Record 8 **Line segment definition**

Field Number	Type	Units	Description
1	Real	cm	Starting X(R) coordinate.
2	Real	cm	Starting Y(Z) coordinate.
3	Real	cm	Final X(R) coordinate.
4	Real	cm	Final Y(Z) coordinate.
5	Real	1/cm	Curvature of the line segment (positive implies centre is to the right of the line from start to end).
6	Real	cm	Bias parameter. This should be equal to the region face bias parameter.
7	Integer	None	Line subdivision. Number of segments the line will be divided into (for best results use line segments that correspond with region faces and have the same subdivision).

Combining Magnetic and Electric Fields

The space charge beam analysis program can be run with a combination of electric and magnetic fields. The following procedure should be used.

1. Set up a model which represents both the magnetic and electric parts of the system.
2. SET the problem type to MAGNETIC, the solution type to one of the vector potential options and the units for magnetic quantities.
3. Assign the magnetic material properties, excitations and boundary conditions.
4. WRITE a data file and solve the magnetic problem with the statics analysis program.

5. Post process the statics solution: use the EXTRA sub-command, TABLE, to create tables of nodal values of the magnetic flux density. The tables should be called XBFLUX and YBFLUX (XY symmetry) or RBFLUX and ZBFLUX (axisymmetry).
6. Remove the magnetic excitations and boundary conditions.
7. SET the problem type to ELECTRIC, the solution type to V (scalar) and the units for electric quantities.
8. Assign electric material properties, excitations and boundary conditions.
9. READ the tables of XBFLUX and YBFLUX.
10. WRITE the data file for the space charge beam analysis. Include the names of the magnetic flux density tables on the WRITE command.
11. Solve the space charge beam problem.

Note that the mesh for the magnetic model must contain all the geometric data that will subsequently be used for the electric field model. If the regions required for the electric field model are entered first, the additional regions that are added subsequently to create the magnetic field model may be erased after operation 5 in the above procedure.

Preparing an OPERA-2d/SP run

The data required to analyse the model is input using the SOLVE command (see Reference Manual for details). The user can set the following information:

- Iterations Data

  This defines data to control the iterative procedure. Values to be set are:

  – Maximum number of iterations

    The value entered sets a limit on the maximum number of non-linear iterations.

  – Convergence tolerance

    This is the convergence tolerance to be applied to the relative change in the electric potential solution.

  – Under-relaxation factor

    An initial under relaxation factor is required to help the non-linear iterations to converge. This should be approximately equal to the ratio of the expected space charge limited current to the maximum (thermally limited) current from the emitters.

- Scale factors

  The scaling factor multiplies non-zero boundary condition values. A list of scaling factors can be supplied.

- ‘New solution’ or Restart

  This is only available if the *.op2* file already contains a solution. The SP solution program allows a solution to be restarted from the result contained in an existing *sp* file. This is a facility that can be used to minimise the solution times for a series of problems with similar geometry or materials, or to continue a solution for more iterations if it has failed to converge. To use this option the input data file must contain a solution.

The space charge analysis program also requires an emitter data file in addition to the model data file. This must have the same file name as the model data file and a file name extension of *emit*. The emitter file can be created using the pre and post processor EMIT command or can be created using a text editor using the format described in the previous section (“Emitter Data File” on page 5-64).

Post Processing

The space charge density calculated by the analysis program is stored in the solution file as a table with the name RHO (in DENSITY units). This is read into the pre and post processor and the distribution can then be graphed or contoured using RHO as a system variable.

The VIEW command can be used to view the sample beams that were used by the analysis programs to calculate the space charge effects. The beam trajectories are stored in a file with the extension *tracks* and the same name as the data file.

Thermal Analysis (TH and THTR)

The Thermal Analysis Programs (OPERA-2d/TH and OPERA-2d/THTR) solve for steady-state and transient temperature distributions. It is possible to apply fixed temperature constraints to nodes and/or faces and cooling conditions to faces. Heat sources are provided as element power densities which can be calculated from an earlier electromagnetic solution.

The model is defined with the OPERA-2d pre and post processor. If analysis of electromagnetically generated heat is required, the electromagnetic analysis must be performed first in the normal way. Before thermal analysis, the user must provide additional information as detailed below. After solution the pre and post processor is used to examine the results. Temperatures throughout the whole model are available. All the usual post processing facilities may be used. Magnetic and thermal results can be displayed simultaneously.

The Equations Solved

The steady-state program (TH) solves the Poisson equation in temperature, T (equation (5.87)) and the transient program (THTR) solves the diffusion equation in temperature, T (equation (5.87)):

$$\nabla \cdot k\nabla T = Q \tag{5.87}$$

$$\nabla \cdot k\nabla T + \rho c\frac{\partial T}{\partial t} = Q \tag{5.88}$$

where k is the thermal conductivity, ρ the density, c the specific heat capacity and Q the input power density.

Boundary Conditions

Three types of boundary condition are available: fixed temperature, perfect insulation and general thermal flow condition. Conditions are applied (after meshing) by using the EXTRA sub-commands CURSOR, FACE or NODE.

The fixed temperature condition, $T=c$, must be supplied on at least one node in order to achieve a solution. It is defined by: **`TEMP t`**

The perfect insulation condition, $\frac{\partial T}{\partial n} = 0$ is the 'natural' boundary condition and thus applies to any boundaries not otherwise set.

The thermal flow condition allows a combination of a constant heat flux, q, and convective heat transfer defined by a heat transfer coefficient, α and the coolant temperature, T_0. The condition is given by

$$-k\frac{\partial T}{\partial n} = q + \alpha(T - T_0) \tag{5.89}$$

where k is the thermal conductivity of the material (see page 5-71). It is on faces defined by: **`THER q a t0`**

Initial Conditions

For the transient analysis (THTR), initial temperatures throughout the model can be specified using a table, otherwise the initial temperature will be assumed to be zero everywhere.

The table is made with the EXTRA sub-command TABLE. The table type must be NODAL, the name must be TEMP and the COMPONENT should be set to the value of the temperature required. The table name must be specified when writing the data file.

Material Properties

Material properties are specified by using the EXTRA sub-command MATERIAL. The material type THERMAL should be chosen. Two values can then be entered for the thermal conductivity tensor and one for the material angle relative to the global x-axis. Transient analysis also needs values for the specific heat capacity and density.

Air elements are usually required for magnetic analysis, but serve no purpose during thermal analysis. The Thermal Analysis Program recognizes air elements and omits them from the equation assembly and solution processes, thus allowing the same model and mesh to be used for both magnetic and thermal analyses. Any material of type NULL or for which material properties are not provided is also treated as air.

Heat Sources

The EXTRA sub-command TABLE may be used after a successful magnetic analysis to store the computed values of the power densities, so that these may be used to calculate the heat sources in the elements for input to the Thermal Analysis Program. For example,

```
TABL COMP=J**2/SIGMA NUMB=1 NAME=HEAT TYPE=ELEM,
UNIT=POWEU/LENGU**3
```

The values stored (specified by **TYPE=ELEM**) are the power densities calculated at the element centroids. Any valid expression for power density can be used. If different expressions are needed in different parts of the model **OPTION=ADD** can be used to add values to an existing table.

The table name must be specified when writing the data file.

Preparing a Thermal Analysis run

The Thermal Analysis module (TH) does not require any additional information in order to be analysed.

The data required to analyse the transient Thermal Analysis module (THTR) is input using the SOLVE command (see Reference Manual for details). The user can set the following information:

- Time Step - Adaptive or Fixed

  The user has a choice to fix the time step at which each solution is calculated, or set an adaptive time step. In Adaptive Time-stepping integration, a time-step variation is defined to achieve a user supplied relative tolerance between successive time-steps. In general, values should be between 0.03 and 0.0001, but values outside this range can be given.

- Output Times

  A list of times at which the solution is stored may be supplied.

Post Processing

Results from the Thermal Analysis Program will have been saved in a file with file name extension *th*. After starting the pre and post processor, this file can be read with

READ *file name*`.th`

All the post processing options are available with the additional variables HEAT and TEMP can be used in expressions for field COMPONENT. In the case of the result of transient analysis, the CASE parameter should be used to choose between the results at different times.

Stress Analysis (SA)

The Stress Analysis Program (OPERA-2d/SA) solves for static stresses and displacements in plane stress, plane strain and axisymmetric conditions. It is possible to apply external constraints to nodes and/or faces; loads due to external forces may be applied, together with Lorentz forces obtained from a magnetic analysis. Various types of material can be specified, including isotropic, orthotropic, transversely isotropic and axi-symmetrically-stratified.

The model is defined with the OPERA-2d pre and post processor. If analysis of magnetic forces is required, the electromagnetic analysis must be performed first in the normal way. Before stress analysis, the user must provide additional information as detailed below. After solution the pre and post processor is used to examine the results. Displacements and stresses throughout the whole model are available. All the usual post processing facilities may be used; in addition the deformed shape of the model can be displayed. Magnetic and structural results can be displayed simultaneously.

The Equations Solved

Each element has its own stiffness matrix and load vector obtained from

$$\int_v [B]^t [C][B] \mathrm{dz} \tag{5.90}$$

and

$$\int_v [N]^t \{X\} \mathrm{dv} + \int_S [N]^t \{T\} \mathrm{dv} \tag{5.91}$$

where

$[B]$	relates nodal strains to nodal displacements according to $\{e\}=[B]\{q\}$
$\{e\}$	is the vector of nodal strains
$\{q\}$	is the vector of nodal displacements
$[C]$	is the matrix of material property constants
$[N]$	is the shape function matrix
$\{X\}$	is the vector of body force densities
$\{T\}$	is the vector of surface tractions

These are assembled into the global matrix equation

$$[K]\{d\} = \{f\} \tag{5.92}$$

where $[K]$ is the global stiffness matrix, $\{d\}$ is the global vector of displacements and $\{f\}$ is the global vector of external forces (obtained by adding external nodal forces to the element load vectors.

After modification to represent the application of constraints to some degrees of freedom this equation is solved for the vector of displacements. Each node has two degrees of freedom in *x* and in *y*. There is a force and a displacement, for each degree of freedom for each node.

Constraints

To enable the global matrix equation to be solved, it is necessary to apply sufficient constraints to the model to eliminate the possibility of rigid body motions in space. We can partition equation (5.92) into the set of unconstrained (u) degrees of freedom, and the set of constrained (c) degrees of freedom:

$$\begin{bmatrix} K_{uu} & K_{uc} \\ K_{cu} & K_{cc} \end{bmatrix} \begin{Bmatrix} d_u \\ d_c \end{Bmatrix} = \begin{Bmatrix} f_u \\ f_c \end{Bmatrix} \tag{5.93}$$

From this, by matrix multiplication, we obtain the two matrix equations

$$K_{uu}d_u + K_{uc}d_c = f_u \tag{5.94}$$

$$K_{cu}d_u + K_{cc}d_c = f_c \tag{5.95}$$

The first of these contains all the information required to solve for the unconstrained set of displacements when rewritten

$$K_{uu}d_u = f_u - K_{uc}d_c \tag{5.96}$$

but to avoid the work necessary to explicitly partition the equations, we solve the matrix equation

$$\begin{bmatrix} K_{uu} & 0 \\ 0 & I \end{bmatrix} \begin{Bmatrix} d_u \\ d_c \end{Bmatrix} = \begin{Bmatrix} f_u - K_{uc}d_c \\ d_c \end{Bmatrix} \tag{5.97}$$

which can be done without any re-arrangement of the degrees of freedom. Note that the unconstrained forces on the right hand side are modified by subtracting terms containing the vector of known displacements. To apply constraint to a node, we specify a displacement for either or both of the node's two degrees of freedom.

Pre Processing

Constraints

Constraints are applied (after meshing) by using the EXTRA command followed by the sub-commands CURSOR, FACE or NODE; having defined a node or face the required constraint values are defined with

CONS *x y*

where numerical values or expressions are substituted for x and y. x or y can be omitted in order to constrain in one direction only. It is possible to constrain nodes to move in directions inclined to the global x axes with

SKEW *angle*

where a numerical value or expression in degrees is substituted for angle. Thus **SKEW 0** is equivalent to **CONS 0**, i.e. movement allowed only in the x direction, and **SKEW 90** is equivalent to **CONS 0**, i.e. movement allowed only in the y direction.

External Forces

Forces are applied (after meshing) by using the EXTRA command followed by the sub-commands CURSOR, FACE or NODE; having defined a node or face the required force values are defined with

LOAD *x y*

where numerical values or expressions are substituted for x and y.

- In XY symmetry, a load on a node is used as a force per unit length. A load on a face is used as a surface traction, i.e. the total force applied per unit length is the load multiplied by the length of the face.
- In axisymmetry, a load on a node is multiplied by $2\pi r$ to give the force. A load on a face is a pressure, and is multiplied by the surface area to give the force.

Material Properties

Material properties are specified by using the EXTRA command followed by the sub-command MATERIAL, which has the parameter OPTION. Refer to Reference Manual for details of the functions provided by OPTION. According to the

material type chosen (ISOTROPIC, ORTHOTROPIC, TRANSVERSELY ISOTROPIC, AXISYMMETRIC-STRATIFIED or NULL) values can be entered for Young's moduli, Poisson's ratios, thermal expansion integrals, etc.

Air elements are usually required for magnetic analysis, but serve no purpose during stress analysis. The Stress Analysis Program recognizes air elements and omits them from the equation assembly and solution processes, thus allowing the same model and mesh to be used for both magnetic and structural analyses. Any material of type NULL or for which material properties are not provided is also treated as air.

Tables of Internal Forces

The command TABLE may be used after a successful magnetic analysis to store the computed values of the force densities, so that these may be used to calculate the body forces on the elements for input to the Stress Analysis Program. For example, in XY symmetry

```
TABL COMP=LX NUMB=1 NAME=XLOAD TYPE=ELEM,
UNIT=FORCU/LENGU**3
TABL COMP=LY NUMB=2 NAME=YLOAD
```

or in AXIsymmetry

```
TABL COMP=LR NUMB=1 NAME=RLOAD TYPE=ELEM,
UNIT=FORCU/LENGU**3
TABL COMP=LZ NUMB=2 NAME=ZLOAD
```

The values stored (specified by **TYPE=ELEM**) are the Lorentz force densities calculated at the element centroids. Any valid expression for force density can be used. If body forces are required in magnetic material, the best results can be obtained using Maxwell stress integrals around each element. This can be achieved using (in XY symmetry)

```
TABL NUMB=1 NAME=XLOAD TYPE=MAXX,
UNIT=FORCU/LENGU**3
TABL NUMB=2 NAME=YLOAD TYPE=MAXY
```

N.B. The COMPONENT is ignored if **TYPE=MAXX** or **TYPE=MAXY**.

Saving the Model

The model region data and mesh information, together with any tables which have been created must be written by, for example

```
WRITE file name XLOAD YLOAD
```

Preparing a Stress Analysis run

The data required to analyse the model is input using the SOLVE command (see Reference Manual for details). The user can set the following information:

- Plane STRESS, plane STRAIN or AXISYMMETRY

Details of each type of analysis are given with the Stress Analysis Example included in the User Guide.

Post Processing

Results from the Stress Analysis Program will have been saved in a file with file name extension *sa*. After starting the pre and post processor, this file can be read with

READ *file name*`.sa`

All the post processing options are available. A useful first check is to ensure that a good solution has been obtained is to plot the deformed shape:

RECO

can be used first to display the undeformed shape, then

CONT +DEFORM VX=DISPX* *xfac* **VY=DISPY*** *yfac*

where suitable numerical values or expressions are substituted for *xfac* and *yfac* displays the deformed mesh; the x and y displacements will be multiplied by these factors to produce a shape which is adequately deformed. Deformed mesh plots can be combined with other types of contours. If an electromagnetic solution was used to provide the input forces for the stress analysis, that solution is still available and can be used in conjunction with the stress analysis solution for further post processing.

More information on preparing, solving and post processing SA problems is given in a separate Application Note.

The DXF Interface Program

The DXF interface program creates a Command Input File from a DXF data file, which can then be read into the OPERA-2d pre and post processor using the $ COMINPUT command. To run the DXF interface program the DXF option should be selected from the top level OPERA-2d menu.

Input Options

The program prompts the user to specify some (or all) of the following information:

- The DXF file name (including the *.dxf* or other extension)
- The Command Input file name (including the *.comi* extension)

The data type POINTS in the DXF file may be represented in the Command Input File in two ways:

- Construction line crosses (**x**) i.e. pairs of construction lines which intersect at the point. If this is chosen, then the size of the cross (the length of the construction lines) may be specified.
- A sequence of construction lines (**l**) i.e. each line connecting an adjacent pair of points.

The program prompts for **x** or **l**.

The data type POLYLINE in the DXF file may be represented in the Command Input File as

- Construction lines (**c**).
- Region boundaries which will be closed to give a Polygonal Region (**r**).
- Regions of **SHAPE=H** based on the Polyline width (**w**). If the Polyline width is 0, construction lines are used.

The program prompts for **c**, **r** or **w**.

SOLID and TRACE data types in the DXF file may be included in the Command input File as regions of **SHAPE=H** in the same way as the Polyline **w** option above. For these regions the number of subdivisions and material number must be given.

Most CAD software will permit subsets of drawings to be stored in a number of DXF files. This is the most effective way of using the interface program since a

number of Command Input Files may be produced with different material numbers, which are read sequentially to form a model.

Adaptive Analysis

Adaptive Analysis is available with the ST, AC and VL solvers. During Adaptive Analysis the value of error in the original solution is used to determine which elements should be subdivided in order to achieve a better solution. Since this routine modifies the mesh itself, any regularly meshed quadrilaterals (shape H and Q regions) are automatically converted to polygons before the analysis starts.

The adaption process is terminated for one of the following reasons:

- Number of refinement iterations required is reached.
- Maximum number of elements specified by the user is exceeded.
- Maximum number of elements allowed by the program is exceeded.
- The accuracy required on the solution is satisfied.

Running Adaptive Analysis

Adaptive Analysis (Mesh Refinement) can be switched on within the ST, AC and VL Analysis options. The following outlines the options which need to be checked:

- Maximum number of iterations

  If zero is entered, the default is set to one iteration; i.e. the process provides the first pre-adaption solution.

- Maximum number of elements

  The refinement routine will increase the discretisation in areas where the solution error is high. The maximum number of elements is the limit on the mesh refinement process.

- Final convergence accuracy

  This is the accuracy required on the refined model, and is expressed in units %. The default value is 5% and this should be reduced if greater accuracy is sought.

Chapter 6
Application Notes

Flux Linkage Calculations in 2D Solutions

Cartesian Problems

In a model with XY symmetry, the cross section of a long device is analysed. There will be X and Y components of flux density and Z directed currents.

Consider two points in the cross section. The flux per unit length linking lines parallel to the Z direction, through the two points, is simply given by the difference in the vector potential (A_z) between the points.

For a line in the x-direction the flux linked in the x-z plane is given by:

$$\phi = \int_{z=0}^{z=1} \int_{x=a}^{x=b} B_y \, dx \, dz \tag{6.1}$$

Since curl(**A**)=**B**, and since only A_z exists in the 2D analysis, then $B_y = -\frac{dA_x}{dx}$, the flux linkage per unit length is:

$$\begin{aligned} \phi &= -\int_{x=a}^{x=b} \frac{dA_z}{dx} dx \\ &= (A_z(a) - A_z(b)) \end{aligned} \tag{6.2}$$

If a coil is made up of a number of filaments and has a turns density of *n*, then the mean flux linkage is given by:

$$\Phi = \int_{\text{coil area}} n\phi \, dx \, dz \tag{6.3}$$

For a coil with uniform turns density, in terms of the vector potential A_z, the mean flux linkage is:

$$\Phi = \frac{1}{\text{coil area a}} \int_{\text{coil area a}} A_z(a)(dx)dz - \frac{1}{\text{coil area b}} \int_{\text{coil area b}} A_z(b)(dx)dz \tag{6.4}$$

The OPERA-2d commands to do this would be:

```
$CONS #Aa area-a
$CONS #Ab area-b
INTA REG1=condb REG2=condb COMP=POT/#Ab
$CONS #POTB INTEGRAL
INTA REG1=conda REG2=conda COMP=POT/#Aa
$CONS #POTA INTEGRAL
$CONS #FLUX #POTA-#POTB
```

where **conda** and **condb** are the region numbers of the conductors, and **area-a** and **area-b** are the region areas, to be inserted as appropriate.

Axi-symmetric Problems

For an axi-symmetric problem, solved in 2-dimensions, the flux linking a loop radius *a* is:

$$\begin{aligned} \phi &= \int_{\theta=0}^{\theta=2\pi} \int_{r=0}^{r=a} B_z r \, dr \, d\theta \\ &= 2\pi \int_0^a B_z r \, dr \end{aligned} \tag{6.5}$$

Since curl(**A**)=**B**, and only A_θ is present (i.e. $\int_0^a B_z r \, dr = r A_\theta$), then:

$$\phi = 2\pi r \, A_\theta(a) \tag{6.6}$$

If a coil is made up of a number of filaments and has a turn density of *n*, then the mean flux linkage is given by:

$$\Phi = \int_{\text{coil area}} n\phi \, dr \, dz \tag{6.7}$$

For a coil with uniform turns density, in terms of the vector potential A_θ, the mean flux linkage is:

$$\Phi = \frac{1}{\text{coil area}} \int_{\text{coil area}} 2\pi r A_\theta \, dr \, dz \tag{6.8}$$

The OPERA-2d commands to do this would be:

1. With an axisymmetric solution, where the solution type is VECTOR POTENTIAL (i.e. A_θ)

   ```
   INTA REG1=cond REG2=cond COMP=2*PI*R*POT/#A
   ```

2. With an axisymmetric solution where the solution type is modified rA (i.e. $r*A_\theta$)

   ```
   INTA REG1=cond REG2=cond COMP=2*PI*POT/#A
   ```

Where `cond` is the region number of the conductor, and #A is the area.

Note that the solution type modified rA gives the best results for all types of axisymmetric geometry.

Particle Trajectory Intersections in OPERA-2d

The following describes the sequence to create trajectory intersections with a line:

Display the model solution as required.

Select the tracks file using

FIELDS↓
Trajectories → **Display** → **Select Track File**

and select the appropriate file.

Display the trajectories on the model using

FIELDS↓
Trajectories → **Display** → **Display trajectories**

Define the intersection line. The local coordinates and appropriate angles should be chosen

FIELDS↓
Trajectories → **Display** → **Display intersections...options**

and then **Intersection line** and define this intersection line by its end coordinates.

To plot a graph of current density crossing the intersection line, select **XY default** or **AXI default** followed by **Display intersections**.

External Circuits in OPERA-2d

Introduction

Many electromagnetic devices are operated while connected to a voltage driven electric circuit. For dc operation it is usually quite simple to determine the current that will be carried by the coils, as this will only depend on the resistance of the winding and any external resistive circuit elements. However, under time varying conditions, the inductance of the coil and external circuit may also be significant and can be time dependent due to magnetic saturation and eddy currents. Consequently, the current in the coil will not be known. To allow problems of this nature to be solved OPERA-2d supports external voltage driven circuits in the AC, TR and RM solvers.

A simple external circuit

Figure 6.1 shows the circuit diagram for the simplest combined external circuit and finite element problem that can be solved in OPERA-2d.

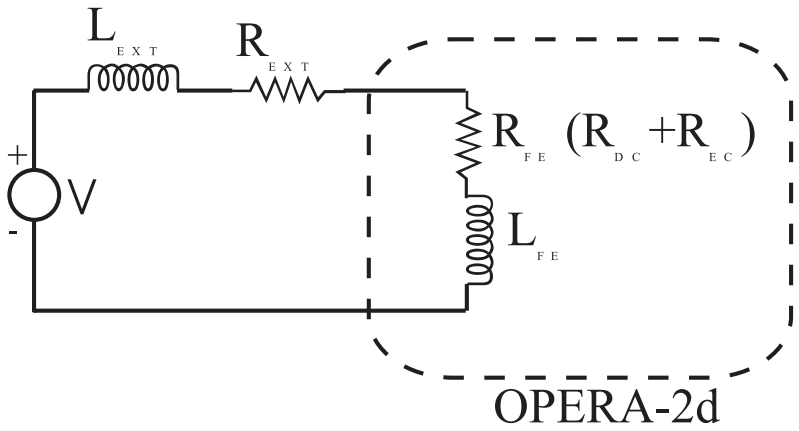


Figure 6.1 Simple external circuit

The voltage supply, V, is connected to the circuit in the OPERA-2d model via a series external resistance, R_{EXT}, and inductance, L_{EXT}. A series capacitance, C_{EXT}, may also be included in the circuit. Specifying zero capacitance implies that no capacitor exists rather than an open circuit. The circuit (or winding) in the OPERA-2d model is also composed of a series resistance, R_{FE}, and inductance, L_{FE}. In the simplest case, the circuit is considered as being constructed from fila-

mentary wires such that skin and proximity effects in the turns are ignored. Windings, where these effects are important, are discussed later in this document. Consequently, R_{FE} has two components – R_{DC}, the DC resistance of the winding, and R_{EC}, the equivalent resistance of the eddy current circuits in the problem referred to the series circuit. The user specifies the values for V, R_{EXT}, L_{EXT} and R_{DC}, while the program computes the values of L_{FE} and R_{EC}.

Circuit With a "GO" Conductor

Figure 6.2 shows the XY cross-section for a simple example of the "GO" conductor of a multi-turn coil above a conducting plate. The boundary condition at X = 0 of tangential flux, implies that the "RETURN" conductor of the coil and the return path for the eddy currents are in the symmetric image. If this coil is connected to a step function voltage, the current in the coil will not rise immediately to its DC value due to the inductance of the winding and the eddy currents induced in the plate. The rise time can be determined using an external circuit with the transient (TR) solver.

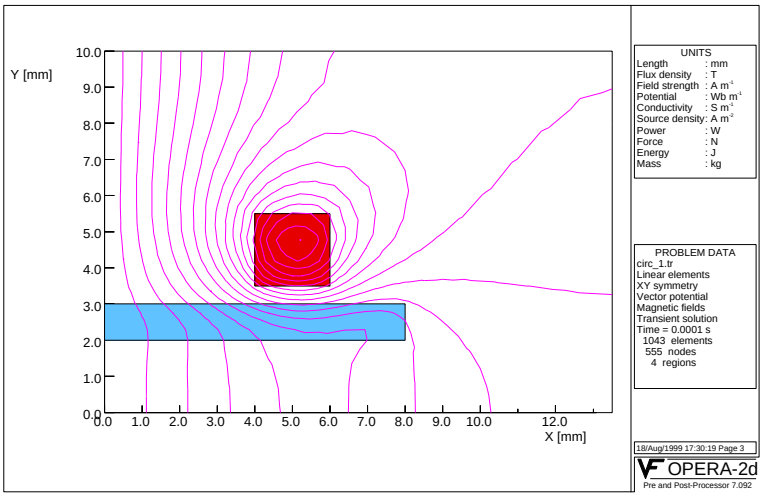


Figure 6.2 Coil above a conducting plate

The model is created as for a normal current driven problem with two exceptions: the current density in the coil is assigned to zero, and the coil region is assigned a conductor number achieved through the

MODEL ↓

Draw Regions → **...conductor data** → **Circuit label**

menu. For this model, the coil region is assigned conductor number 1. After mesh generation, the user defines the external circuit through the

MODEL ↓
Circuits → Define circuit

menu. Global parameters for the circuit are specified first via a table to be completed by checking options or supplying values.

Element	Value	Explanation
Circuit type	Filamentary	Skin and proximity effects are not included
Symmetry	2	Circuit currents return outside the model
Length (XY only)	50	The coil is 50 mm long in the Z direction
Peak voltage	1.5	Voltage applied to circuit, *V*
Phase angle	0	Used in AC solutions only
Resistance	0.02	Value of R_{EXT} in Ω
Inductance	0	Value of L_{EXT} in Henrys
Capacitance	0	Value of C_{EXT} in Farads (0 = no capacitance)
Initial voltage	0	Voltage on capacitance at time = 0 sec. for TR/RM solutions (in Volts)

Following this, the program requests the elements (conductor numbers) that build up the circuit. In this model, only one conductor is specified.

Item	Value	Explanation
Conductor number	1	The next series element to be included in the circuit
Sense of conductor	GO	The current will flow into the plane of the model
Number of turns	150	The number of *series* turns in this conductor
Resistance/unit length	0.0012	Resistance of the *wire* used to construct the coil (=1.2 mΩ/mm)

The circuit is terminated by quitting from the input table. The values in these tables determine the value of R_{DC} and the number of turns and length of the coil is used to compute L_{FE} and R_{EC}.

The circuit can be listed, and facilities are available to make corrections. When the circuit has been entered correctly, the *.op2* file for the transient analysis is created specifying a step function drive for circuit 1.

Figure 6.3 shows the exponential rise of the current in the winding, displayed using the new **`FILE` → `Graph data in file`** facility. Note that it approaches its dc value, V / (R_{EXT} + R_{DC}) = 83.2 mA.

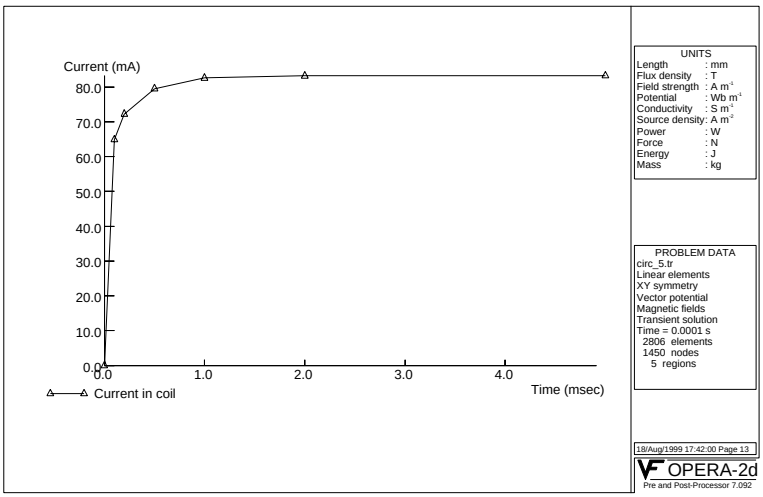


Figure 6.3 Transient rise of current in coil from step voltage input

Circuits with “GO” and “RETURN” conductors

In a model where there is no symmetry, it is necessary to include both the “GO” and “RETURN” conductors in the circuit. Figure 6.4 shows a modification to the above problem with the coil positioned asymmetrically above the plate.

The “GO” part of the circuit is assigned conductor number 1 and the “RETURN” conductor number 2. The external circuit is defined similarly as for the symmetric

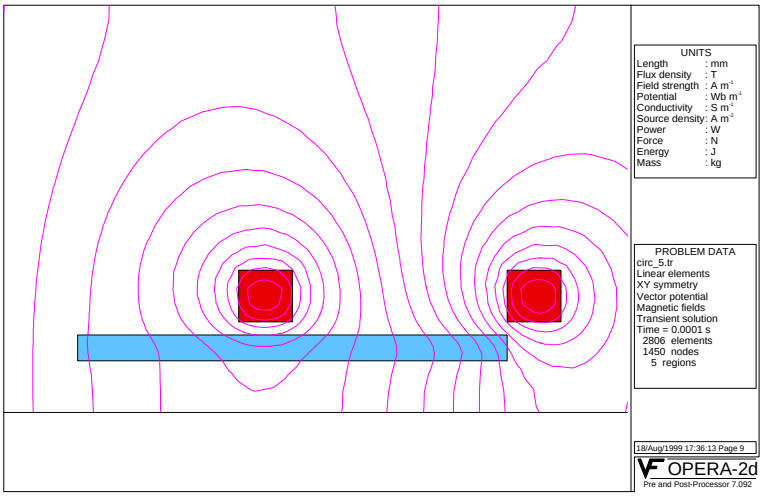


Figure 6.4 Asymmetric model

case except that “Symmetry” is assigned to “1”, since no currents return outside the model, and a second element must now be added to the circuit.

Item	Value	Explanation
Conductor number	2	The next element to be included in the circuit.
Sense of conductor	RETURN	The current will flow out of the plane of the model
Number of turns	150	The number of *series* turns in this conductor
Resistance/unit length	0.0012	Resistance of the *wire* used to construct the coil (=1.2 mΩ/mm)

Using “GO” and “RETURN” conductors in this way, a complex winding (for example, a distributed phase winding in an electrical machine) can be built up from a number of elements. Parallel conductors can also be included by assigning the same conductor number to different regions of the model.

“Bulk” conductors in external circuits

The previous examples used filamentary conductors – skin and proximity effects in the turns were ignored and the current density in each turn of the coil is assumed to be uniform. Some coils and circuits are constructed from more substantial pieces of conducting material where these effects may become important – for example in large electrical machines or resistive accelerator magnets. Even

windings that can be considered filamentary under “normal” operation can exhibit these effects if operated at very high frequency or with very fast transients. Consequently, it can be important to include these effects in the voltage driven circuit. This modifies the simple circuit defined above such that R_{DC}, which can be computed from the user defined circuit parameters, is replaced by R_{EM}, the “dynamic resistance” which is computed from the finite element solution and includes the redistribution of the current within the conductor.

Figure 6.5 shows the asymmetric plate problem now driven from a four turn coil with each turn having a 0.4 x 0.4 mm cross-section. In this example, the voltage source is an AC supply at 10 kHz. As the skin depth of copper at this frequency is about 0.7 mm, some redistribution of the current in the coil is seen, shown by the coloured contours in the turns. Consequently, $R_{EM} \times R_{DC}$ and can be evaluated from the power dissipation and current in the conductor.

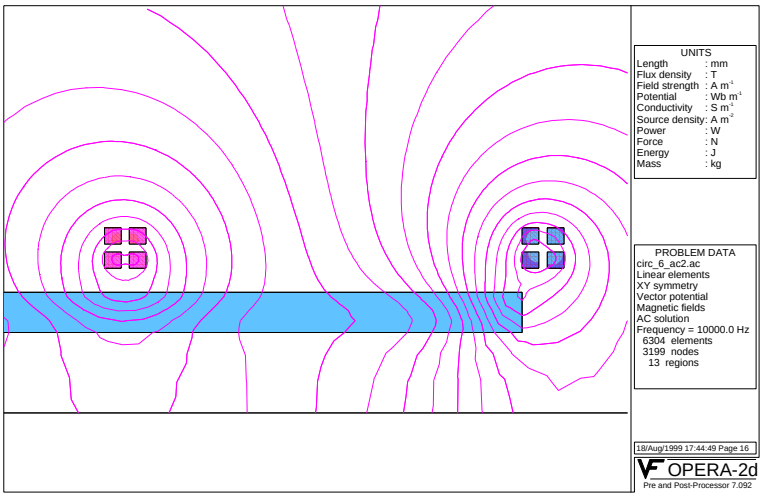


Figure 6.5 Asymmetric coil with bulk conductors

Setting up a problem with bulk conductors is similar to filamentary conductors. Each region (or regions) representing the GO or RETURN conductor of each turn is assigned a separate conductor number. Hence, in the above example, the conductors have been numbered 1 through 8. The conductivity of each region is also specified, as this will be used to determine the skin and proximity effects. When the circuit is defined, the global parameter for “`Circuit type`” is set to “`Eddy Current`”. The elements of the circuit are again defined using the conductor numbers and “GO” or “RETURN”, but the number of turns and resistance/unit length are not requested. It is assumed that the conductor represents a single turn and the resistance, R_{EM}, is determined within the finite element solution from the conductivity of the material and its volume.

Accurate Results for Stress Analysis

Introduction

Many electromagnetic devices introduce significant internal forces in rigid magnetic or conducting bodies. To determine the effects of these forces OPERA-2d includes SA, the Stress Analysis Program, which solves for static stresses and displacements in plane stress, plane strain and axisymmetry. Forces may be determined from the electromagnetic analysis, but the module may also be used with user-defined loads. The purpose of this note is to give guidance on how to obtain the best results with the SA module.

The Finite Element Mesh

The Finite Element mesh consists of triangular elements with linear or quadratic shape functions. A linear shape function assumes that the displacements at any point in the element can be expressed as a linear function of the nodal displacements. The quadratic shape function uses a quadratic function interpolation of the nodal displacements at any point in the element.

Like all Finite Element programs, the stress analysis solver is sensitive to the element aspect ratio. The aspect ratio is defined as the ratio of the largest to the smallest side of the triangular element. Elements of high aspect ratios must be avoided. The aspect ratio should ideally be 1, although aspect ratios up to 10 have been proven to perform well. The problems caused by higher aspect ratios relate to the elements difficulty to represent local changes in stress and strain. Ill-conditioned matrices may also result, which lead to a completely incorrect set of displacements. With quadratic elements, oscillatory behaviour in the field representation results.

Boundary conditions

External constraints such as zero displacement in X or Y (R or Z in axisymmetry) or both can be applied to nodes or faces of a region, hence facilitating the modelling of common stress analysis problems such as a simply supported beam or a cantilever. In addition, different types of loading can be assigned to a beam. Point loads are represented as a load applied to a single node, while distributed loads are applied by selecting a face of a region.

The following section will concentrate on typical stress analysis problems. Problems for which simple analytical solutions exist have been chosen for the analysis, to provide a basis for comparison with results from OPERA-2d models of varying discretisation strategies.

Example 1: Simply supported beam with distributed loading

Theory

The simply supported beam is 1m long, 0.1m in thickness and of unit depth. One end of the beam is fixed with a PIN-type connection and the other end is fixed with a ROLLER-type connection. A uniform load *W*/ unit length of 5 N/m is applied along the length of the beam.

The expression for the displacement in the Y-direction is

$$\nu = \frac{W}{12EI}\left(Lx^3 - \frac{x^4}{2} - \frac{L^3 x}{2}\right) \tag{6.9}$$

where:

E is the Young's modulus (in N/m^2)

I is the second moment of mass (in m^4)

L is the length of the beam

x is the distance from the PIN joint end of the beam to the point of measurement

The maximum displacement will occur at the mid-span and is equal to

$$\nu_{max} = \frac{5}{384}\frac{WL^4}{EI} = 2.6\times10^{-8} m \tag{6.10}$$

The second moment of mass is equal to

$$I = \frac{dh^3}{12} = 8.33\times10^{-5} m^4 \tag{6.11}$$

where:

d is the depth of the beam

h is the thickness of the beam

Model

The model has one bottom end node constrained with zero movement in both X and Y to represent the PIN type connection, while the other bottom end node is constrained only in Y, hence representing the ROLLER type connection. A uniform loading was achieved by applying a load of (0,-5) N/m to the top face of the beam, noting that the negative sign signifies a downward Y-directed force. A discretisation of 100×10 (length x thickness) subdivisions was chosen, resulting in a fine and regular mesh of unity element aspect ratio, as shown in Figure 6.6.

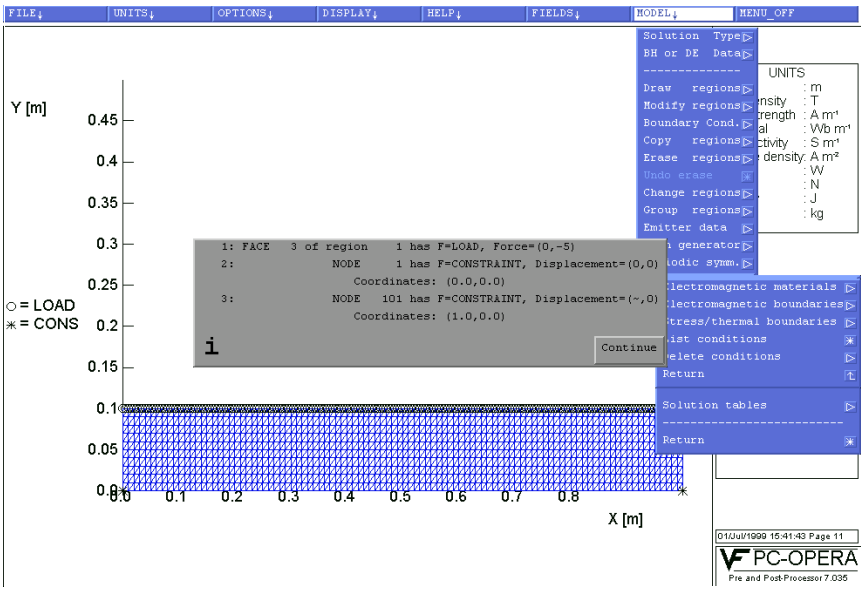


Figure 6.6 Model of a simply supported beam showing regular mesh and boundary condition definition

Solutions

OPERA-2d post processing facilities include deformed shape plots, based on the computed values of the displacements in X and Y (DISPX and DISPY), as illustrated in Figure 6.7. In this model, the maximum value of DISPY occurs at mid-span, as expected, and the computed figure of 2.7×10^{-8}m is in excellent agreement with the value obtained using analytical methods, hence reflecting the quality of the mesh and assignment of correct boundary conditions. There is therefore no need to further refine the mesh.

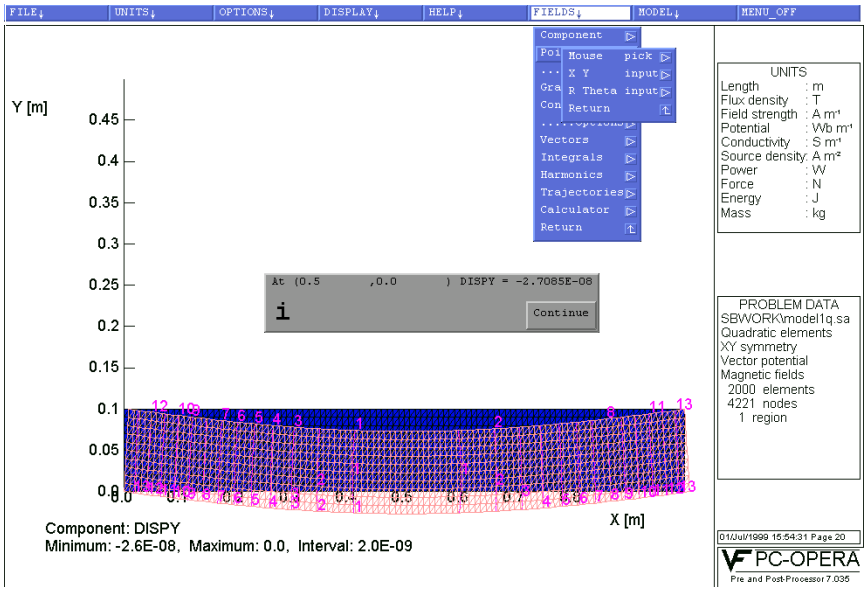


Figure 6.7 Simply supported beam deformation under load showing a maximum deformation at mid-span of 2.7×10^{-8}m

Discussion on Aspect Ratio assignment

It is advisable to always construct regular meshes with aspect ratios ranging from 1 to 10, so as to avoid inaccuracies in the solution. The combination of high aspect ratio geometry and elements, together with the use of quadratic element representation can lead to oscillatory behaviour in field representation results. One such example constructed was a simply supported beam 10 m in length and 0.1 m in thickness, with a discretisation of 10x25, resulting in a fine mesh along the beam thickness but a crude mesh along its length.

In order to examine the Y-directed displacement along this beam, a line graph extending from (0,0) to (10,0) of component DISPY was drawn, as shown in Figure 6.8. The displacement along the beam is seen not to increase smoothly with distance away from the beam supports as expected, and the displacement in the centre of the beam is not in agreement with the analytical predictions.

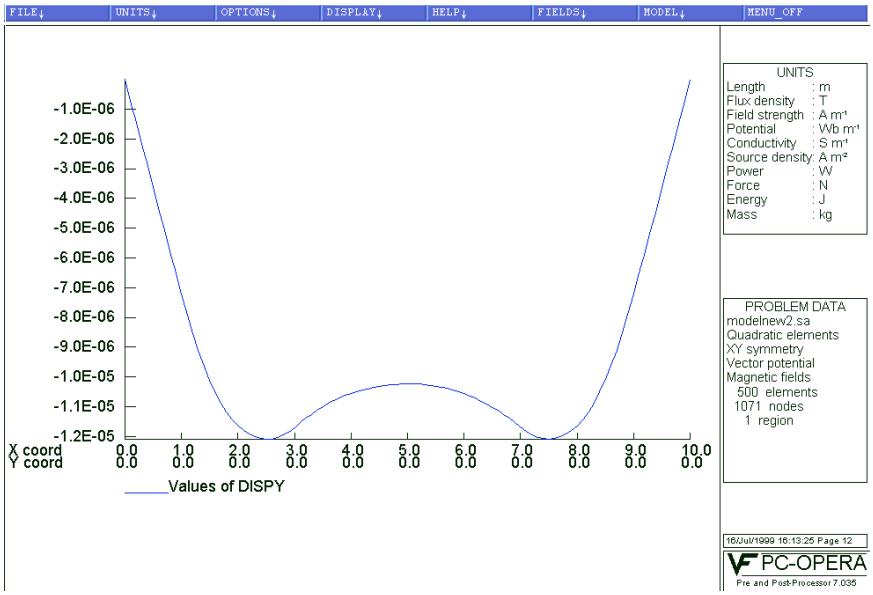


Figure 6.8 Variation of displacement along the length of a simply supported beam computed from a finite element model, showing effect of incorrect aspect ratio elements

Example 2: Cantilever carrying a concentrated load

Theory

The cantilever is 1m long, 0.1m in thickness and of unit depth. A point force of 5N/m is applied at the centre of the cantilever, i.e. 0.5m away from the clamped end.

The deflection at the point of application of load, a distance l from the clamped end, may be expressed as

$$\nu_l = \frac{Wl^3}{3EI} \tag{6.12}$$

resulting in a figure of 8.33×10^{-8}m in this example. The maximum deflection for such a beam may be computed using

$$\nu_{max} = \frac{Wl^2}{2EI}\left(L - \frac{l}{3}\right) = 2.08\times10^{-7} m \tag{6.13}$$

noting the variables:

E the Young's modulus (in N/m^2)

I the second moment of mass (in m^4)

l the distance from the load application point to the clamped side of the beam

L the length of the beam.

Model Preparation

The equivalent model is set up in plane symmetry (plane stress) and has one side clamped i.e. one face of the beam is constrained with zero movement in both X and Y. A regular mesh was constructed with 50×25 subdivisions and a resulting element aspect ratio of 5, as shown in Figure 6.9. A load of (0,–5) N/m was applied at the top centre node of the cantilever model. Linear elements were selected for the first analysis submission.

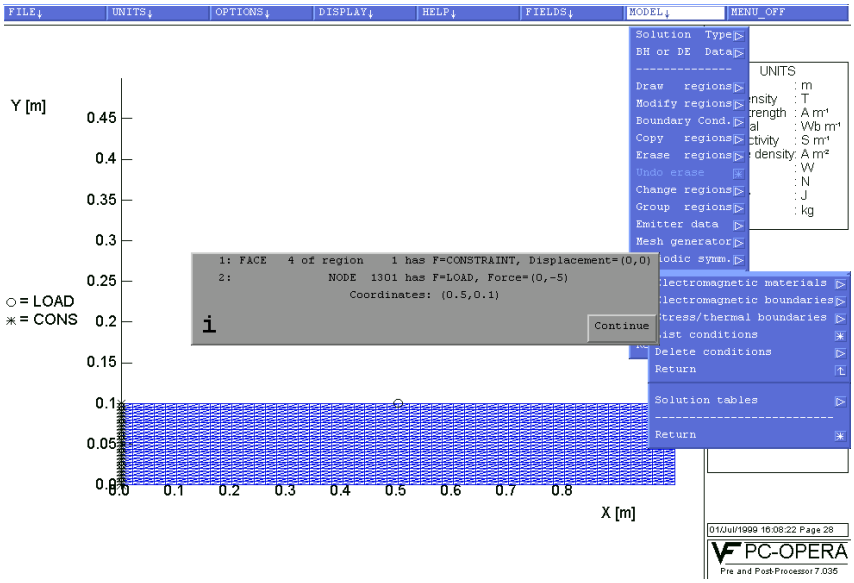


Figure 6.9 OPERA-2d model for cantilever with point load

Solution

The deformed cantilever shape predicted by OPERA-2d is shown in Figure 6.10 with the maximum displacement occurring at the ‘free’ end of the cantilever as anticipated, and reaching 2.0×10^{-7}m. The predicted mid-span displacement is 8.13×10^{-8}m. Both values are in good agreement with analytical predictions.

The results can be further improved by the use of quadratic element representation. The quadratic element solution yielded a maximum displacement of 2.1×10^{-7}m, only marginally exceeding the analytical prediction.

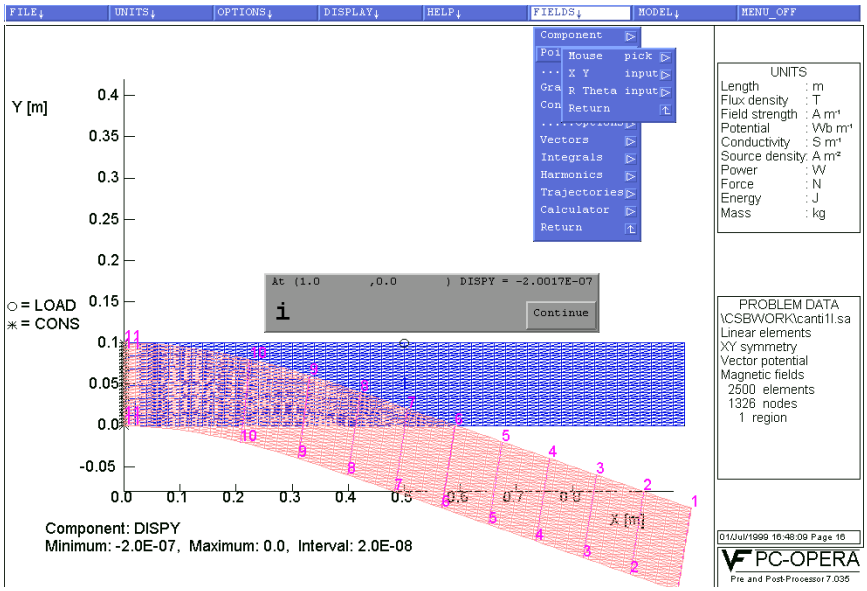


Figure 6.10 Deformed cantilever with displacement contours

Example 3: Thin disk with a clamped periphery and distributed load

Theory

The thin disk has a diameter of 22m and a thickness of 0.2m. Note that the model aspect ratio is high, and therefore care must be taken in defining the mesh. The disk is clamped around the whole of its periphery. A distributed load of 2 N/m^2 is applied to the disk.

The maximum deflection occurs at the centre of the disk and can be expressed as

$$w_{max} = \frac{pa^4}{64D} \tag{6.14}$$

where:

a is the radius of the disk (in m)

p is the pressure applied on the disk (in N/m^2)

D is the flexural rigidity.

The flexural rigidity can be expressed as

$$D = \frac{Eh^3}{12(1-\nu^2)} \tag{6.15}$$

where:

h is the thickness of the plate

v is the poisson's ratio.

Based on the dimensions given in the present example, the maximum deflection value is 2.08×10^{-5}m

The maximum stresses occur at r=a and z=±h/2. They are expressed as

$$\sigma_{r_{max}} = \pm\frac{3}{4}\frac{pa^2}{h^2} = 4537Nm^{-2} \tag{6.16}$$

Model preparation

The model is set up in axisymmetry and has its whole side at R=11 m constrained in both R and Z (this implies an R,Z constraint all around the peripheral surface of the disk). The mesh chosen is of 55×10 subdivisions, yielding an aspect ratio of 10. The axisymmetric model set up is shown in Figure 6.11.

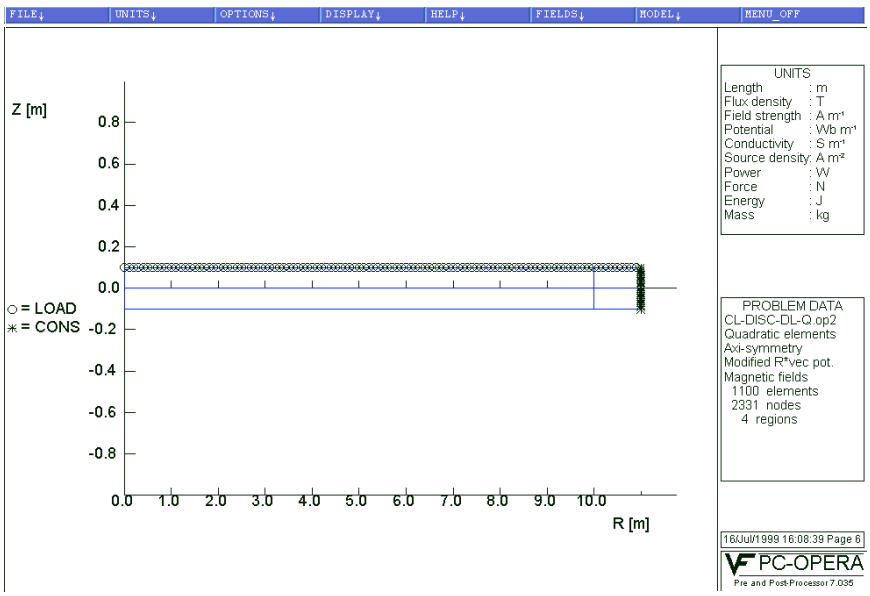


Figure 6.11 Clamped disc model preparation in axisymmetry, showing effect of insufficiently defined boundary conditions

Results

The maximum displacement value returned by OPERA-2d is 2.08×10^{-5}m, in perfect agreement with the analytical prediction. However, the distribution and values of σ_r were wrong. The reason for this is the lack of one boundary condition. The model must be constrained so that the centre axis of the disc (at R=0) does not move along the R-direction, hence creating a 'virtual hole' in the model.

Revised Model

A revised clamped disc model was prepared, with an additional boundary condition imposed, namely an R=0 constraint (i.e. no movement in the R direction) applied along the side of the axis of rotation of the disc. The resulting displacement remained unaltered, while the σ_r distribution was significantly improved. The maximum value of σ_r returned from the revised OPERA-2d model is 4960 N/m^2. This is in reasonable agreement with the theoretical value of 4537 N/m^2 though it can be further improved by refining the mesh.

Electromagnetic and Stress Analysis of an SRM

Introduction

Following the application note “Accurate Results for Stress Analysis” on page 6-11 this note serves to introduce the user to more complex, so called “coupled problems”, where results from an electromagnetic analysis are coupled to the stress analysis module.

The model chosen in this instance is a switched reluctance motor. The doubly salient, deeply saturated structure makes electromagnetic finite element analysis a necessary tool for evaluation of the machine's performance. Stress analysis performed on switched reluctance machines is important, as the pulsed nature of torque production induces vibration (and hence acoustic noise).

This example illustrates the use of the pre and post processor with the statics and stress analysis solvers. The method of transferring electromagnetic analysis results to the stress solver, as applied to an electric motor, is given. Initially the electromagnetic problem is solved. The results are read into the pre and post processor and examined before being transferred with additional mechanical information to a data file for input into the stress analysis solver. The results of this analysis are then examined by further post processing.

Solving the Electromagnetic Problem

The model of a simple 2-phase 4-pole switched reluctance motor, designed for a high speed application, is shown in Figure 6.12. The back-iron width is relatively small compared to the stator pole width and this results in reduced stator stiffness and hence more vibration.

In order to obtain an accurate electromagnetic solution, certain parts of the mesh must be well refined, especially the air gap and the tips of the stator and rotor teeth. The mesh in this region is shown in Figure 6.13.

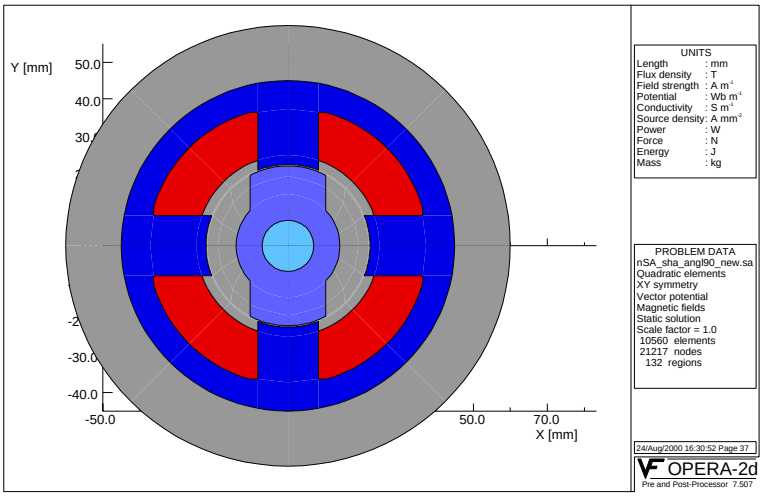


Figure 6.12 A 2-phase switched reluctance motor

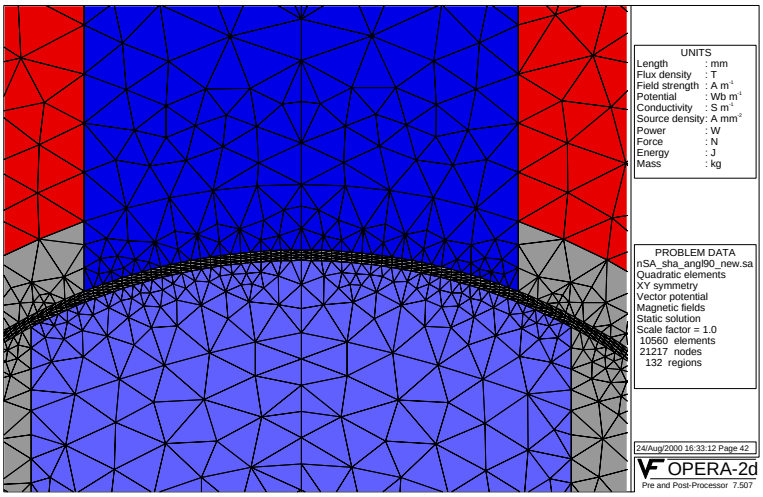


Figure 6.13 Meshing at the pole tips

Examining the solution

The average torque produced by the motor, assuming a constant excitation current (this can only be true at very low speeds) can be computed simply by evaluating the change in co-energy at the 0^o (un-aligned) and 90^o (aligned) rotor positions. In order to obtain the co-energy of the motor at each position, the area integral command (**INTA**) is used, having set the component to **HDB**. The software computes the H.B/2 and H.dB integrals from which the co-energy value can be

extracted. The average torque produced by the motor was estimated to be 0.0525 Nm/mm stack length.

Figure 6.14 illustrates the field distribution at the 45^o position. This is a high torque producing position, as the rotor teeth tend to be pulled into alignment with the excited stator teeth. The instantaneous torque at this position is expected to be much higher than the average torque computed earlier. Instantaneous torque can be readily evaluated be performing an arc integral (**INTC**) in the middle of the air gap, a command which returns the X and Y directed Maxwell stress force and the resulting torque about a specified pivot point. The torque at this position was computed to be 0.139 Nm/mm.

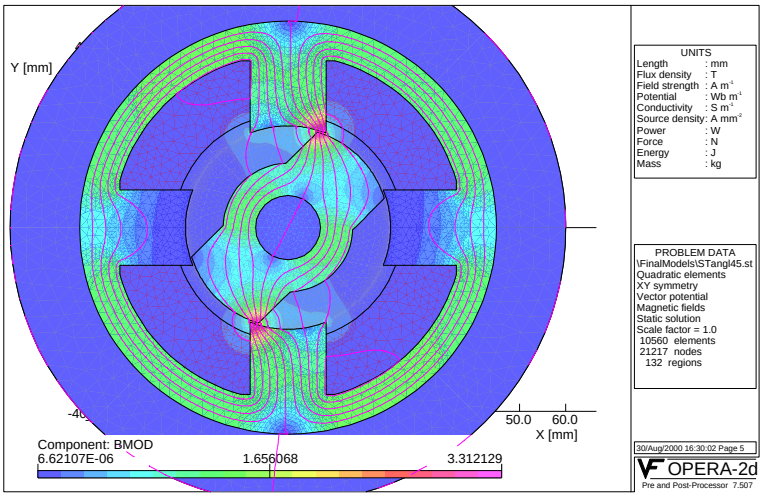


Figure 6.14 Field pattern in a 2-phase switched reluctance motor

The 45^o (overlapping) and 90^o (aligned) positions will be examined further with the stress analysis solver.

Entering the Mechanical Data

In order to mechanically characterise a motor, it is necessary to define the mechanical material data constants for at least three materials, namely the stator, rotor and coils. The following commands were used to define the stator:

```
MODEL ↓
      Extra options → Stress/thermal materials → Define a
                                                 new material

Material number=3.
```

The dialogue boxes with material data were completed as follows:

Material name = **Stator**
Material type = **Isotropic**

followed by Young's modulus = **210e9** and Poisson's ratio = **0.29**

The same properties were assigned to Material Number 4 (**Rotor**) whereas Material Number 1 (**Coils**) was assigned:

Young's modulus = **120e9**
Poisson's ratio = **0.29**

Entering the mechanical boundary conditions

The definition of appropriate boundary conditions really depends on how the motor is fixed to its frame, and therefore many valid combinations of boundary conditions can exist.

In this model, the stator was mechanically fixed at 4 points around the outer edge, behind the centre of each pole. The centre of the shaft was also constrained in both X and Y. The condition was applied by selecting

MODEL ↓
Extra
options → **Extra**
conditions → **Stress/thermal**
boundaries → **Constrained**
in both

with: **Function for X or R = 0**
and: **Function for Y or Z = 0.**

selecting **Pick nodes** and clicking on the 5 points described above.

Alternative boundary conditions (for reference)

The geometry of the machine can sometimes be such that the constraint cannot be expressed simply with movements in X and/or Y, as it could in the present model. In such cases a skew condition could be applied. A skew condition with a specified angle can be interpreted as a constraint on specified points lying on this angle, allowing these points to move radially but not azimuthally. The skew constraint is also an alternative way to constraint points lying on the X or Y axis. For instance, a skew defined as:

```
MODEL ↓
      Extra
      options → Extra
                conditions → Stress/thermal
                             boundaries → Skewed
                                          constraint
```

with: **Angle= 0**

has the same effect as a Constraint in Y (=0) at the same point.

Creating tables for mechanical analysis

The stress analysis module requires data from the electromagnetic solution, relating the loading of the problem to the finite element mesh, to be available as *tables*. The integrals of the Maxwell Stress on each element in the X and Y direction (**XLOAD** and **YLOAD**) were created. As an example, the following created a Table of X-directed forces on each element:

```
MODEL ↓
      Extra
      options → Solution
                tables → Make a
                         new table ...
                              ... Options → Maxwell
                                            stress X or R
```

and **Return**

followed by

```
Make a new table...
      ... Make table
```

completed with

Field component	=	**pot**
Table Number 1	=	**1**
Name	=	**xload**
X-derivative	=	
Y-derivative	=	
Unit expression	=	**FORCU/LENGU**3**

Accept Dismiss

Note that the field component entry is actually irrelevant in this case, due to the previous selection of the x-component of Maxwell Stress.

Having all data in place, the *.op2* file must be written to include the geometry, mechanical materials and boundary conditions as well as the Tables of the Maxwell Stress Integrals. This is done by selecting

FILE ↓
Write File → Analysis Data

selecting the SA solver and opting for Plain Stress. Plain Stress assumes that the lamination is a thin sheet which is not prevented from contracting in the normal - Z - direction. Plain Strain would assume an infinite thickness, where each plane parallel to the paper would be constrained to remain plane. Essentially, neither of these two cases exist in isolation, so the calculation could be repeated with the alternative choice, and the 'worst case scenario' used for design calculations. Subsequently,

FILE ↓
Write File → Write Model

typing a suitable filename and clicking on the XLOAD and YLOAD Tables to include them in the *.op2* file.

Examining the Stress Analysis Results

Displaying the results as a deformed mesh

The excited stator teeth act as magnet shoes, pulling the rotor into a position of minimum reluctance. As the rotor moves towards alignment, forces of attraction are acting on the excited stator and rotor teeth. In the aligned position, the two bodies would still tend to close the gap that separates them, in an attempt to minimise the reluctance of the magnetic circuit. The rotor is therefore in tension while the stator is in compression.

The deformation due to loading may be viewed using the deformed mesh feature. To do this, the regions which need to be examined must be selected, and the displacement vectors scaled. The following shows the options selected for a stator deformation plot:

FIELDS ↓
Contour plot...Options

then

Select material

specifying **Material Number 3**.

followed by

Standard plot *(toggles to* **Deformed plot)**
Deform options

The parameter box was completed by setting the x and y components to **DISPX*1e9** and **DISPY*1e9** respectively. The multiplier used is based on the minimum and maximum values of the displacement. Selecting

FIELDS ↓
Contour plot → **Label style** → **No labels**

and

No refresh *(toggles to* **Refresh)**

followed by

Execute

produces the result shown in Figure 6.15.

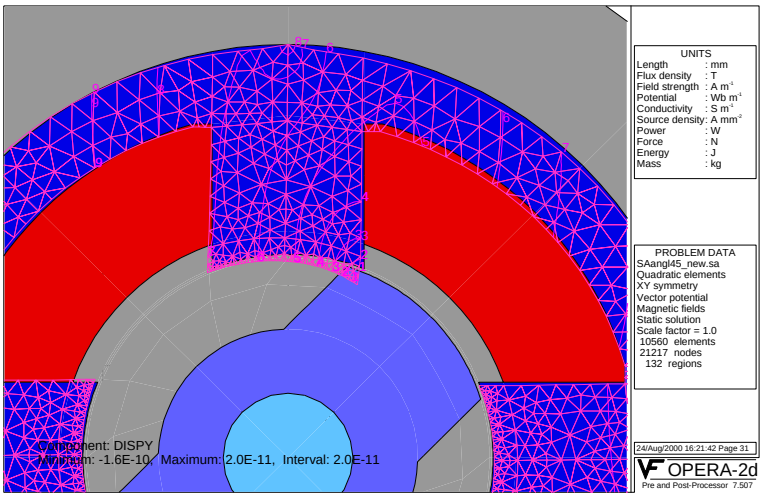


Figure 6.15 Stator displacement (not to scale)

A similar plot can be obtained for the rotor (refer to Figure 6.16 below).

A contour plot of the values of displacement can also be plotted. Figure 6.17 illustrates the values of DISPY in the aligned position, confirming the trends described earlier: DISPY is negative in the stator (indicating a downward movement) and positive in the rotor.

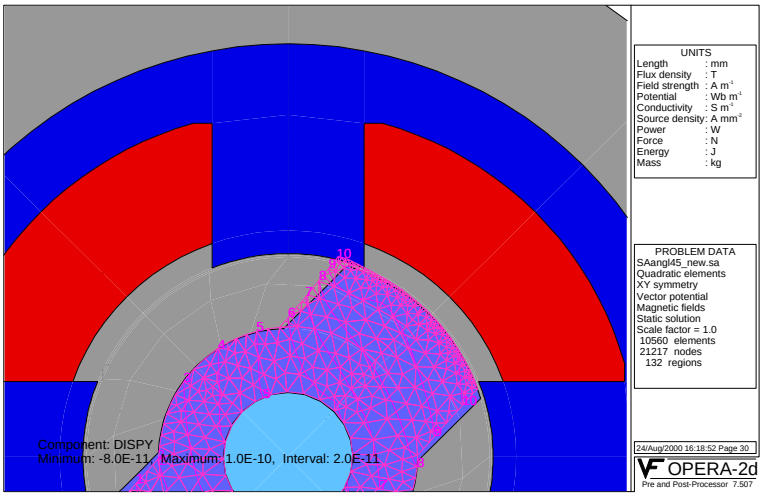


Figure 6.16 Rotor displacement (not to scale)

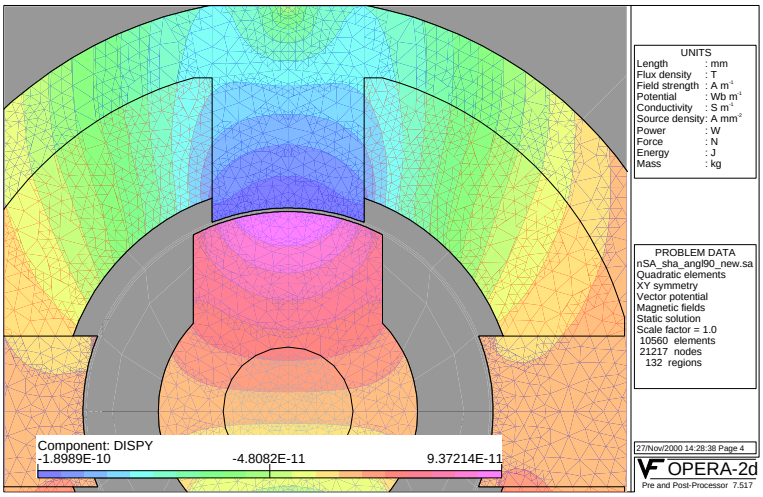


Figure 6.17 Contours of stator and rotor displacement in the aligned position

Conclusion

This Application Note has shown how OPERA can be used not only for the full electromagnetic analysis of a switched reluctance motor, but through close coupling of the ST and SA solver, also assessing the stiffness of the motor structure and characterising its deformation behaviour under load.

Power and Energy Calculation in AC Solutions

Power Calculations

If a current and voltage waveforms vary as:

$$\begin{aligned} i &= I\ sin(\omega t) \\ v &= V\ sin(\omega t) \end{aligned} \tag{6.17}$$

then the power in the system is given by:

$$\begin{aligned} P(\omega t) &= vi = VI sin(\omega t) sin(\omega t - \beta) \\ &= VI[\,cos(\phi) - cos(2\omega t - \phi)] \\ &= A - \frac{VI}{2} cos(2\omega t - \phi) \\ &= A - \frac{VI}{2} cos(\phi) cos(2\omega t) - \frac{VI}{2} sin(\phi) sin(2\omega t) \\ &= A + B cos(2\omega t) + C sin(2\omega t) \end{aligned} \tag{6.18}$$

Hence when ωt=0, then

$$\begin{aligned} P(0) &= A + B \\ P\left(\frac{\pi}{4}\right) &= A + C \\ P\left(\frac{\pi}{2}\right) &= A - B \end{aligned} \tag{6.19}$$

The time average power is given by:

$$P_{tav} = \frac{P(0) + P\left(\frac{\pi}{2}\right)}{2} = A = \frac{VI}{2} cos(\phi) \tag{6.20}$$

Energy Calculations

Since non-linear materials are not modelled, in a magnetic circuit, **B** and **H** are sinusoidal. Replacing the *i* and *v* with **B** and **H**, the same arguments follow, such that the time average energy is given by:

$$\mathbf{E}_{tav} = \frac{\mathbf{E}(0) + \mathbf{E}\left(\frac{\pi}{2}\right)}{2} = A = \frac{\mathbf{B} \cdot \mathbf{H}}{2} cos(\phi) \qquad (6.21)$$

General

P is the system variable POWER, and *E* is the system variable ENERGY, set after the INTA command, and ω*t* is set by the TIME parameter. Further application notes are available under the relevant commands in the Reference Manual.

Inductance Calculations in OPERA-2d

In general, with non-linear materials where permeability is a function of field strength, the inductance of a device will also be a non-linear function of excitation. The flux linking a single coil carrying a current I becomes:

$$\Phi(I) = \frac{1}{N}\int_{i=0}^{I} L(i)di \qquad (6.22)$$

(neglecting hysteresis effects), where Φ is the flux linking the coil, L is the inductance, and N is the number of turns in the coil. .

Single Coil Problems - Linear Materials

In problems where there is only one coil, and no permanent magnets or external driving fields exist, inductance can be calculated from the stored energy in the problem. The energy is calculated using the area integral over all regions in the model, using the command: **`INTA REG1=1 REG2=*.`**

The inductance is computed using the expression (where E is the energy and i is the current in the coil):

$$E = \frac{1}{2}Li^2 \qquad (6.23)$$

Multiple Coils – Linear Materials

Multiple coils make inductance calculations using energy much harder. Instead, another method of calculating inductance uses the flux linking the coils (see for example “Flux Linkage Calculations in 2D Solutions” on page 6-1).

The simplest method of calculating the flux is to calculate the difference in magnetic vector potential between one point in the “GO” section, and one in the “RETURN” section of the coil.

A better method would be to generate an average vector potential over both sections of the coil, using the area integral with component set to POT, and dividing by the area of the region. The flux would then be the difference between these values in the two sections.

Using the flux, Φ, and the current in the coil, I, the self inductance can be calculated as:

$$L = \frac{N\Phi}{I} \tag{6.24}$$

A similar equation can be used for the mutual inductance, using the flux linking one coil due to the flux generated by the current in another. This is calculated using:

$$M_{ij} = \frac{N_i\Phi_i}{I_j} \tag{6.25}$$

Using the solution to the model with one coil switched on will allow you to find the self inductance of the coil, and the mutual inductance of other coils with respect to that coil.

Multiple Coils – Non-linear Materials

If you wish to use non-linear properties it is necessary to have a solution with all coils at normal operating conditions. This gives the equation for the flux linking coil 1 as:

$$N_1\Phi_1 = L_1I_1 + M_{12}I_2 + M_{13}I_3 + \ldots + M_{1n}I_n \tag{6.26}$$

for a model with n-coils. Similar equations can be generated for Φ_2 etc. Note that all the inductances will be functions of the currents.

The model must then be altered so that the current in any one of the coils is changed by a small amount (ΔI), saving the model, and generating a new solution file (using restart run to speed matters up).

The small change should not greatly affect the field, but a small change of $\Delta\Phi_I$ will occur for each coil. For example, for a change of ΔI_1:

$$\begin{aligned} N_1(\Phi_1 + \Delta\Phi_1) &= L_1(I_1 + \Delta I_1) + M_{12}I_2 + M_{13}I_3 + \ldots + M_{1n}I_n \\ N_2(\Phi_2 + \Delta\Phi_2) &= L_2I_2 + M_{21}(I_1 + \Delta I_1) + M_{23}I_3 + \ldots + M_{2n}I_n \\ &\ldots \end{aligned} \tag{6.27}$$

Hence taking the difference between equation 6.27a and equation 6.26, we can calculate L_1:

$$L_1 = \frac{N_1\Delta\Phi_1}{\Delta I_1} \tag{6.28}$$

Equation 6.27b and equation 6.26 will give M_{21}, and so forth. Modifying I_2 would allow similar calculations for L_2 etc. Therefore, finding the self and mutual inductance for all *n* coils requires n models to be solved.

This method can also be used if other energy sources are present, e.g. permanent magnets or external driving fields, as it works on the basis of a *change* in field.

A change in energy calculation is possible but would require more solutions to be calculated, as only one piece of information is found from each model.

Use of Command Scripts to Calculate Fourier Series

It is possible to generate command scripts that will generate the different components of a Fourier series. These commands scripts make use of the $ commands specified within the Reference manuals and are applicable to both 2D and 3D solutions.

The following is an example *.comi* file that calculates the Fourier components of POTENTIAL where the full period has been modelled. The commands are those for OPERA-2d but can be easily transferred to 3D by using the LINE command followed by a PLOT command within the main $ DO loop.

```
/ Set the number of Fourier components
$CONS #n 8
/ Set the line for the integral - for a straight line
$CONS #x1 0
$CONS #y1 0
$CONS #x2 10
$CONS #y2 0
/ Assume the line of integration covers
/ one whole period.
/ If the line only covers half a period -
/ set #symm to 2 and remove the odd or even
/ component from the fourier series (see later)
$CONS #symm 1
/ Get the length of the line and the fractional
/ distance along the line
/ If working on e.g. an arc section,
/ #T should be the total Angle subtended by the arc,
/ #X the fractional distance around this arc
$CONS #T SQRT((#x2-#x1)**2+(#y2-#y1)**2)
$PARA #X SQRT((x-#x1)**2+(y-#y1)**2)/(#T*#symm)
/ Set the function to be analysed
$PARA #Fx POT
/ Set the component that will be integrated for each
/ component
/ Temporarily set #i
$CONS #i 0
$PARA #Cosi Cos(2*PI*#I*#x)
$PARA #Sini Sin(2*PI*#I*#x)
/ Loop through each component
$DO #I 0 #n
/ Get the Cos Fourier components of the function #Fx
/ Not necessary if the function is Odd
INTL #x1 #y1 #x2 #y2 ERRO=128 COMP=(2/#T)*#Fx*#Cosi
$CONS #A%int(#I) INTEGRAL
/ Get the Sin Fourier components of the function #Fy
/ Not necessary if the function is Even
INTL #x1 #y1 #x2 #y2 ERRO=128 COMP=(2/#T)*#Fx*#Sini
$CONS #B%int(#I) INTEGRAL
$END DO
/ If #Fx is an Even function
$CONS #A0 #A0/2
```

The results of this analysis will be the components of the Fourier series described by

$$F(x) = a_0 + a_1 cos(2\pi xt) + a_2 cos(4\pi xt) + \ldots + b_1 sin(2\pi xt) + b_2 sin(4\pi xt) + \ldots \quad (6.29)$$

or using the notation of the *.comi* file:

```
F(x) = #A0 + #A1*Cos(2*PI*#x/#T) + #A2*Cos(4*PI*#x/#T) + ...
... + #B1*Sin(2*PI*#x/#T) + #B2*Sin(4*PI*#x/#T) + ...
```

The *.comi* file above is a general purpose command file and must be modified for use in the correct context. If only part of the model has been created within the software, the commands above must be changed to reflect the partial nature of the results of the integral command, i.e.

F(x) is an Odd Function: An = 0

F(x) is an Even Function: Bn = 0

Only 1/4 period: Even harmonics e.g. **`A2`**, **`A4`**, **`B2`**, **`B4`** are zero.

Complex Material Properties

Introduction

The introduction of complex material properties in a.c. solutions can be a powerful tool for modelling the physical behaviour of a material. The effect of a complex material property is to allow a phase difference between the two vector quantities in the constitutive relationships:

$$\mathbf{B} = \mu \mathbf{H}$$
$$\mathbf{D} = \varepsilon \mathbf{E}$$
$$\mathbf{J} = \sigma \mathbf{E}$$

The most commonly used of these is a complex μ as a model for hysteresis. This has the effect of giving an elliptical relationship between magnetic flux density, **B**, and magnetic field strength, **H**. For example, a material specified with a relative permeability of 100 and a phase angle of 20 degrees ($\mu_r = 100\, e^{-j(\pi / 9)}$) gives a loop as shown. Complex permeability is available in OPERA-2d/AC.

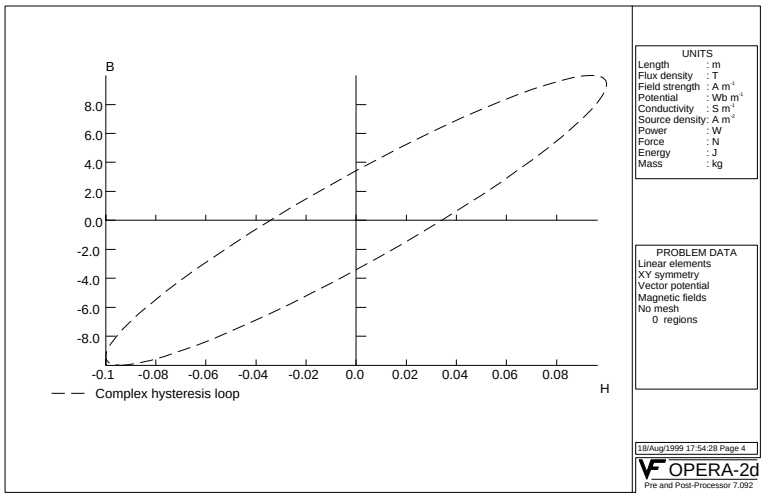


Figure 6.18 Typical hysteresis loop

Accessing Complex Permeability in OPERA-2d

There are two steps that must be taken to use complex μ in the OPERA-2d AC solver. The first step is that the phase angle of the material must be defined. This

is achieved in the BHDATA command, where the PHASE parameter specifies the phase angle in degrees. This is found under

MODEL↓
BH or DE Data → Complex phase lag

The second step is to request that the complex material properties are invoked in the ac analysis. In keyboard mode, this is achieved under the SOLVE command with the sub-command: **DATA +CMU**

Alternatively, switching complex permeability on and off is found as one of the options on the ac analysis options menu, which is presented when Steady state harmonic (AC) is chosen using:

FILE ↓
Write file → Analysis data menus

Complex permeability in non-linear problems

When complex permeability is used in non-linear materials, the choice of phase angle can become difficult. As shown in Figure 6.19, choosing a single phase angle that is applied over the whole **B** vs. **H** characteristic can give very different sized hysteresis loops. In fact, the hysteresis losses (and hence loop size) for a material operating at a peak flux density of 20 kGauss and at 25 kGauss will be nearly identical.

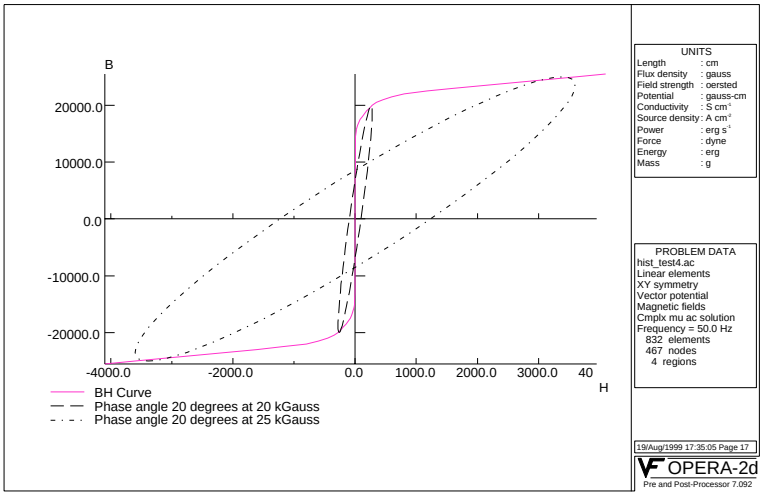


Figure 6.19 Using different phase angles for different parts of the BH curve

To obtain a more accurate solution, it is better to use several different phase angles for different operating regimes on the curve. This can be achieved as follows.

1. Run the model non-linearly without complex permeability
2. Determine the variation of peak flux density in the magnetic materials and discretise into a number of ranges. The shape of the **B** vs. **H** characteristic will dictate this discretisation but it is unlikely that all ranges will be equal. For example, the ranges (0 – 0.5), (0.5 – 1.0), (1.0 – 1.2), (1.2 – 1.3), (1.3 – 1.35) and (1.35 – 1.4) Tesla may be suitable for a material that is not very saturated.
3. Re-assign the materials in the model so that a different material number is used for the parts of the model that fall into each range of flux density.
4. Run the new model non-linearly with complex permeability using the same **B** vs. **H** characteristic for all materials but assigning a different phase angle for each material number.

Modelling of Asynchronous (Induction) Motors

Introduction

OPERA-2d and, in particular, the AC and RM solvers have been successfully implemented for the modelling of asynchronous (induction) motors. The AC solver is capable of accurately modelling the currents induced in the cage rotor, resulting from the rotating 3-phase magnetic field generated in the stator. The induction motor performance can be characterised at a range of rotor speeds and/or stator winding excitation levels. Typical motor performance parameters such as the variation of torque with slip frequency, induced rotor currents and associated power losses are readily computed in OPERA-2d AC. In addition, the Rotating Machines (RM) Solver, a transient eddy current solver, can be used to model the rotational effects, hence predicting transient effects at start-up and torque ripple.

OPERA-2d AC Model Preparation

In induction motors, the relative velocity between the rotor and the field on the stator is called slip, **s**, and torque production is heavily dependent on it. The difference between the rate of rotation of the stator-produced magnetic field and the rotor is called slip speed and effectively, this can be thought of as the rotational frequency at which the stator-produced rotating field cuts the rotor bars. This phenomenon can be successfully modelled in OPERA-2d, by setting the exciting stator field to rotate at the slip frequency, rather than the true AC frequency

Current or voltage driven motors can be set up in OPERA-2d. The preparation of a current driven problem can perhaps require some pre-modelling calculations on the user's part. Typically, the user will hold data on the rated current and winding configuration (star or delta, number of coils per phase, number of phases in each slot etc.) and must translate that to user-defined conductor regions of designated current densities. Current excitation is catered for within the Region Material Properties. Voltage driven coils can be defined using the **`External Circuit`** options.

External Circuit data consists of power supply characteristics (voltage, phase), a series external lumped resistor, inductor and capacitor (which may or may not be associated to internal power supply characteristics), the length of the circuit and the OPERA model conductors (i.e. stator coils) which the circuit is connected to. Conductors attached to external circuits are appropriately labelled using the **`Circuit Label`** option. Lumped resistance and inductance values (external to the

FEA model) can be included in the Circuit definition to account for elements not catered for in the two-dimensional model, such as the stator end winding resistance and leakage inductance. Rotor bar connections can also be modelled in a similar fashion. Rotor bars shorted by an end ring are assigned the same circuit label and the `total current fixed` option is used.

In the AC code, the use of the slip frequency requires further adjustments to be made to any external circuits which may be attached to the motor. The use of the slip frequency to drive the model results in values for variables such as the back-emf which are reduced by the ratio of slip frequency (**fslip**) to synchronous frequency (**fsync**). This can be remedied by scaling the length of the circuit (in the External Circuits section). However, the resistance per unit length must then also be adjusted, such that the true circuit resistance value is maintained. End region inductance figures will also need to be adjusted for solutions at different speeds. The following is an example of such a calculation for a 4-pole, 60 Hz induction motor:

Motor Data
Synchronous speed = 1800 rpm (equivalent to 60 Hz) Rotor speed = 1740 rpm Slip speed = 60 rpm Slip frequency = 2 Hz
Data for a solution at starting (fslip=fsync)
R/mm = 1.4815 E-05 Ohms/mm (true value) Machine length = 95 mm End winding length = 270 mm End winding resistance (lumped value) = 0.004 Ohms Hence, Resistance / turn = (1.4815 E-05 * 95) + 0.004 = 0.0068 Ohms
Data for a solution at 1740 rpm
Machine length = 95 * 60 / 2 = 2850 mm(back-emf adjustment) R/mm = 1.4515 E-05 * 2 / 60 = 0.04938 Ohms/mm Hence, Resistance / turn = 0.0068 Ohms
Note: Lumped resistor values need no adjustment.

Figure 6.20 illustrates the mesh of a 4 pole induction motor. Fine discretisation is applied to the stator and rotor pole tips as well as the air-gap, where significant MMF is dropped.

The air-gap of the machine should have at least 3 layers of elements and should preferably be constructed from H or Q type regions. Figure 6.21 ‘homes in’ on the gap region of a 6-pole, deep slot fractional slot winding induction motor. The aluminium rotor bars are also finely discretised, in order to correctly capture the induced eddy currents.

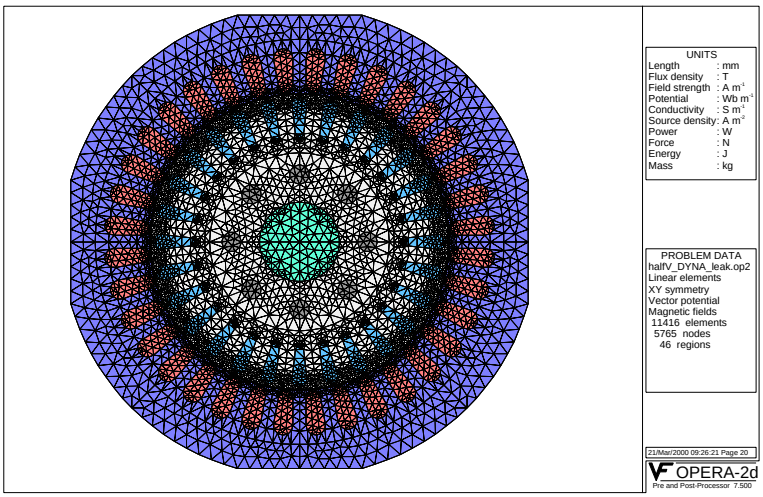


Figure 6.20 A 4-pole induction motor model

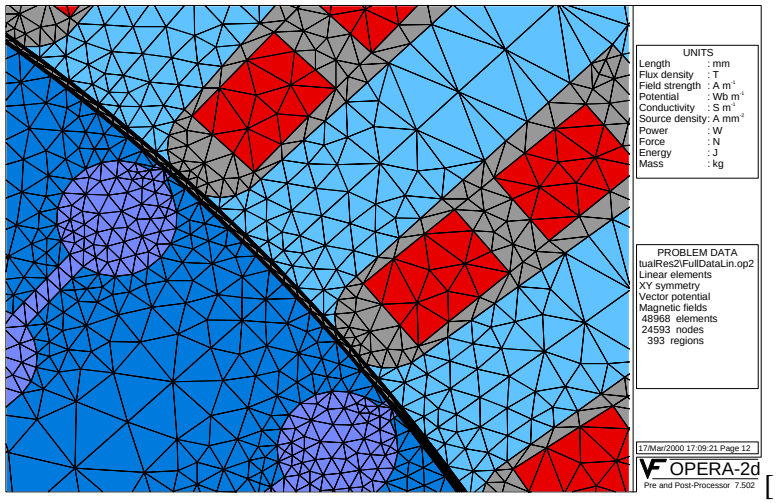


Figure 6.21 Mesh discretisation in the air-gap of a 6-pole machine

Sample Post Processing facilities for OPERA 2d AC Solutions

Figure 6.22 illustrates a plot of vector potential in a 4-pole induction motor, from which the user can readily identify that the correct 4-pole field has been set up by the Circuits definition.

The current density in the aluminium slots can also be plotted, and the current flowing down each slot can be evaluated by performing a surface integral of the current density over the selected aluminium regions.

The forces acting on the rotor, as well as torque, can be computed accurately with special built-in functions such as the Maxwell Stress Integral. The following command computes the X-Y force components acting on the surface of the rotor, as well as the resulting torque around the specified pivot point.

```
INTC TIME=TAVERAGE

INTC RAD=59.8, P1=0, P2=360, XC=0, YC=0, ERR=5000
```

giving the following output:

```
Integration from 0.0 to 360.0       around (0.0        ,0.0        )
 Radius: 59.8       ,   Number of steps:  5000
 Steady State Ac TIME=TAVERAGE
 Integration accumulator set to zero.
 Integral of x-component of Maxwell stress           = -2.3225E-04
 Integral of y-component of Maxwell stress           = 4.81375E-04
 Integral of torque around (0.0        ,0.0        ) = 203.0109637
 Integral of POT                                     = 0.0
```

The line integral describes a circle around the rotor, half-way along the air-gap between the stator and the rotor. The resulting torque figure of 203 Nm/m, which for the present machine of 95 mm translates to 19.285 Nm. This figure is very close to the machine rated torque of 20.88 Nm. The discrepancy is attributed to end winding inductance effects which cannot be modelled completely with two-dimensional software.

In order to compute the current loss in the rotor circuits the following commands are entered:

```
CONT COMP=J**2/SIGMA,

INTAREA TIME=TAVERAGE,

INTA REG1=40,REG2=40
```

Resulting in the following report:

```
Integration over region   40
 Steady State AC TIME=TAVE
 Total current (integral J ds)                  = 0
 Integral of potential (integral A ds)          = 0
 Stored energy/unit length (integral A.J/2 ds)  = 3.3027E-14
 Stored energy/unit length (integral B.H/2 ds)  = 0.00157132
 Power/unit length (integral (J**2)/sigma ds)   = 1.22667
 Force/unit length (integral JxB ds)             = (1.037E-07,8.157E-07)
```

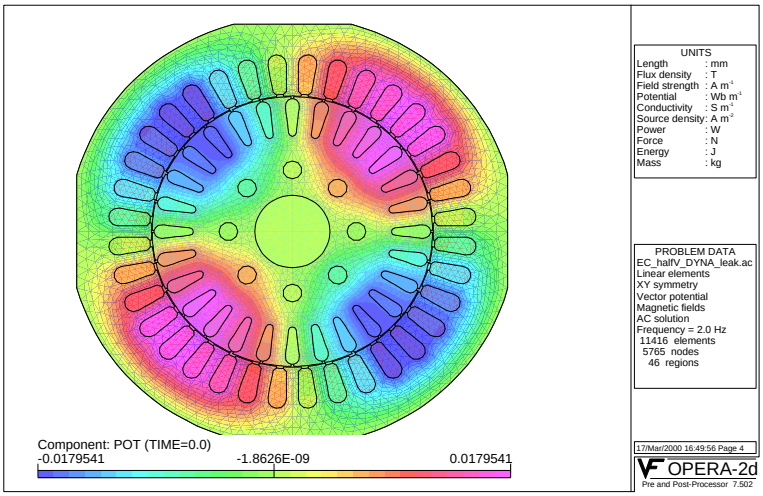


Figure 6.22 Vector Potential variation in the 4-pole induction motor

```
Integral of J**2/SIGMA                          = 1.22667E+09
OK
```

Note that here, the component setting to **`J**2/SIGMA`** was superfluous, and was shown for illustration purposes. The integral of **`J**2/SIGMA`** is one of the pre-set integrations carried out by the program each time INTA is executed. The software returns a value of 1.22667 W/mm which, for the present machine length of 95 mm, translates to 116 W.

Obtaining OPERA 2d RM Solutions

Transient effects such us the build-up of current in the machine under a locked rotor can be modelled using the transient solver TR. A current or voltage waveform can be defined, and the field distribution at different times examined. The complete motor performance characterisation, including the effects of rigid body rotation, can be realised by the use of the Transient Rotating Machines Solver RM. Certain guidelines in the model preparation are to be observed when the RM solver is to be employed:

1. The RM Code requires an RM Air Gap Region. A single Air Gap region is defined automatically by the software, following a user prompt which must include the average radius inside the RM Air Gap.
2. If model symmetry is exploited, the **`symmetry number`** (2 for a 180 degree model, 4 for a 90 degree model, -4 for a 90 degree model with negative symmetry) is entered in the RM Region Gap data. The software will automatically apply the appropriate boundary conditions.

3. When using the RM Code, the true value of frequency of the exciting stator field must be used. The speed of the rotor can also be entered as a separate parameter.
4. The use of the true exciting frequency for the stator windings renders unnecessary any adjustments made to the Circuits definition.

The RM Code is a transient analysis Code. Initial machine transients brought about when a current surge or a voltage is applied at the terminals of the motor can therefore be successfully examined. The examination of the complete machine response as a function of time, up to the steady state operation requires the running of the motor model under examination for a sufficient time period, which is time constant dependent. The specification of many output times has little effect on the computation speed though it does result in a much larger solution file.

The length of the time step which the software uses between each solutions is crucial to good convergence and computation times. In general, it is difficult for the user to assess the correct time step for any model. It is therefore considered good practice to choose the ‘adaptive time step’ option, especially in the early stages of simulation. The software continuously reports the length of the time steps it is using as it progresses through the solution, and these can be recorded by the user.

As the software converges on the use of a (relatively) fixed time step, the user can stop and restart (or run further simulations) using the time step previously observed. It is important to stop the simulation soon after an ‘output time’, so that no computing time is wasted.

Current Diffusion From a Superconductor

This application note describes how to use OPERA-2d to model the diffusion of current from superconducting filaments to the surrounding resistive material.

Physical description of the model

When a superconducting filament becomes resistive, the current will spread from the filament to the surrounding resistive material. This can be simulated by solving the diffusion equation, in order to model eddy currents that flow to resist the change in the field distribution.

A transient OPERA-2d analysis can be used to model the effects of a change in material conductivity. An initial solution must first be created with the current flowing in the filament. A restart case can then defined and a transient analysis performed which will model the diffusion of current from the filament.

Example

The following figures show the results from the analysis of a single conductor in a surrounding aluminium resistive material. A fixed current initially flows in the conductor, see Figure 6.23, and the conductor becomes resistive at a time of 1 second. The figures show the diffusion of current after that time, the total current remains constant.

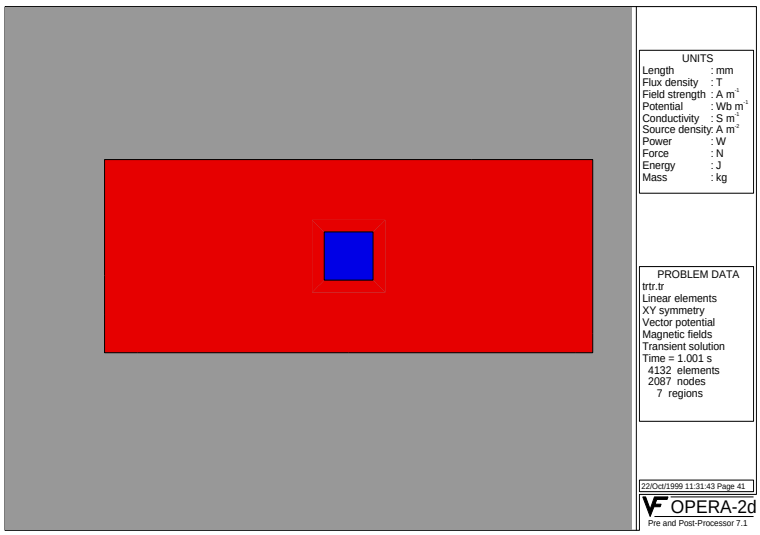


Figure 6.23 The model - Aluminium (red), Conductor (blue)

Setting up the models

OPERA-2d does not expect a restart calculation to be performed using a different conductivity, in addition it cannot solve for a DC starting case where the superconductor has zero conductivity. The following steps are therefore required to set up the models:

Initial solution

The superconductor is modelled with a high conductivity compared to the aluminium. In the example shown above the conductivity of the aluminium was 10^{11} S/m and the superconductor was initially defined with a conductivity of 10^{15} S/m. The regions representing the conductor and aluminium were given a conductor label and the integral current constraint was applied to them all so that the total current will remain constant (region parameters **`N=1 SYMM=1`**). The integral current constraint is applied through the GUI with the following path:

```
Model ↓
      Modify
      regions → Modify
                region → Pick
                         region → Conductor
                                  data → Total current
                                         unlimited
```

which will then toggle to **`Total current fixed`**.

The current density in the conductor must be **`asc`** and in the aluminium it must be **`asa`** where **`a`** is a constant, chosen to give the required total current (this is a restriction applied by OPERA-2d to conductors in the same symmetry group with an integral current constraint, the voltage required to drive the current in the different materials must be the same).

The model is then stored for transient analysis, with a DC drive and a single output time point at 1 second, and the solution calculated by running the transient solver.

Creating the model to examine current diffusion

The initial solution can then be read into the OPERA-2d pre processor. The conductivity of the superconductor regions should be changed to its resistive value. The current density for both the aluminium and conductor regions must be adjusted to give the same total current, used for the initial solution, and with the correct ratio between the values required for the new value of conductivity, for example

```
$CONSTANT #Aa  Cross_sectional_area_of_Al
$CONSTANT #Ac  Cross_sectional_area_of_Sc
$CONSTANT #Sa   Conductivity_of_Al
$CONSTANT #Sc   New_Conductivity_of_Sc
$CONSTANT #I     Required_total_current
$CONSTANT #JA #I/(#Aa+#Ac*#Sc/#Sa)
$CONSTANT #JC #JA*#Sc/#Sa
MODI REG1=1 REG2=1 DENS=#JC SIGMA=#Sc    ! updating the superconductor
MODI REG1=2 REG2=6 DENS=#JA             ! updating the Al regions
```

Modifying the data removed the solution, this can be restored by typing

```
READ CASE=1
```

This is recognised as a special option in OPERA-2d, only the solution is read from the model that was last used. The data should now be stored for a transient, restart analysis with DC drive and time output from 1 second onwards (or whatever time was used for the initial solution).

Why does this work

The restart transient solution starts from a field and current distribution that came from the conductivity values used for the initial solution. This is not the steady state current distribution that corresponds to the new values of conductivity, the diffusion of the current towards these steady state values will be calculated.

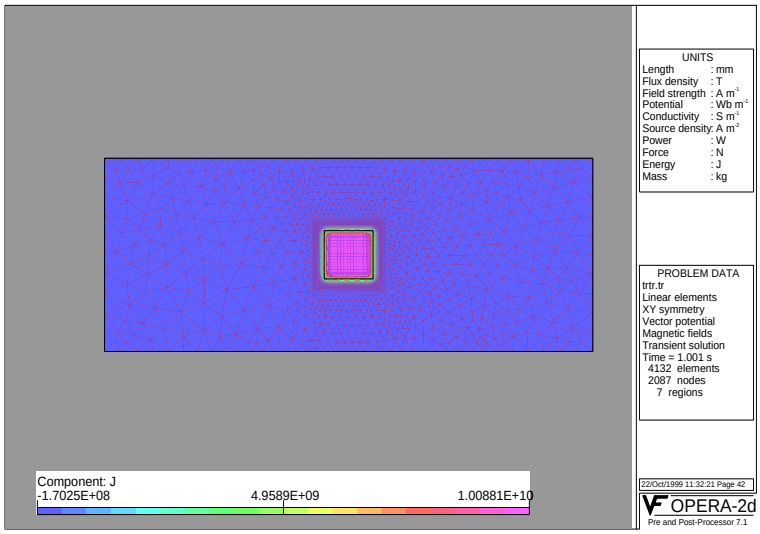


Figure 6.24 Contours of current density at t = 1.001 seconds

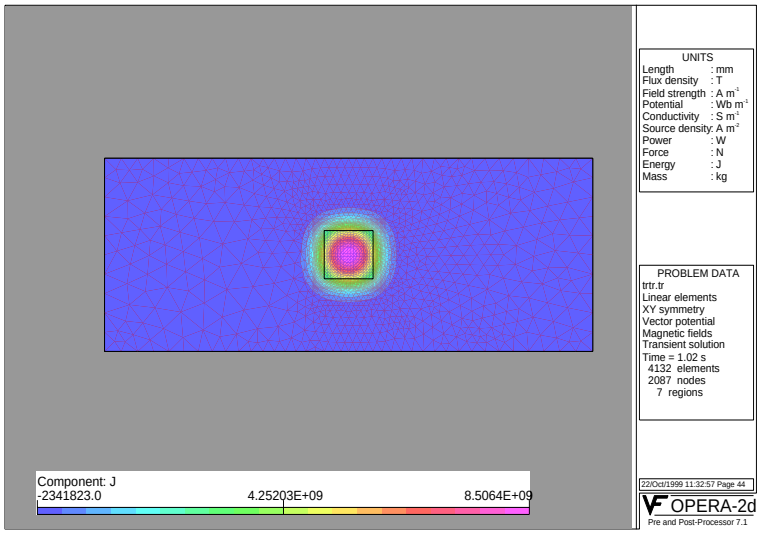


Figure 6.25 Contours of current density at t = 1.02 seconds

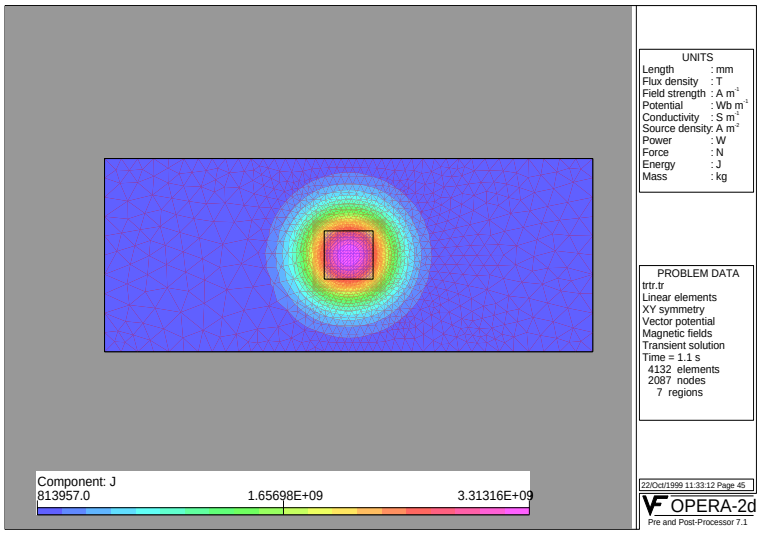


Figure 6.26 Contours of current density at t = 1.1 seconds

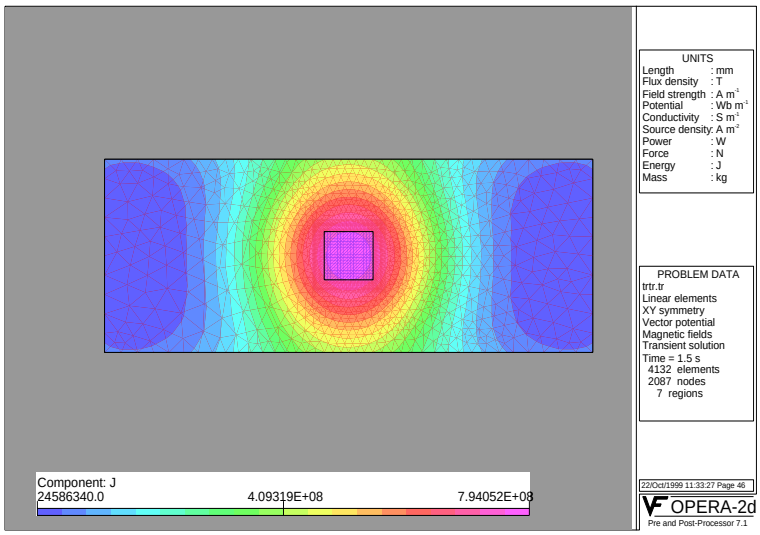


Figure 6.27 Contours of current density at t = 1.5 seconds

Plasma Free Surface Emitter in 2D

This example is showing some features of the plasma emission model in OPERA-2d/SP. It's a simple structure with a few controlling electrodes and an anode. There is a rotational symmetry, and therefore it can be modelled in 2D axi-symmetry.

Figure 6.28 shows the simple structure. The emitter is in the lower part of the model, and the electrons are travelling upwards towards the anode. The trajectories are calculated only in free space. As soon as a particle hits any material, it will stop. In most cases the anode is being used to stop the beam.

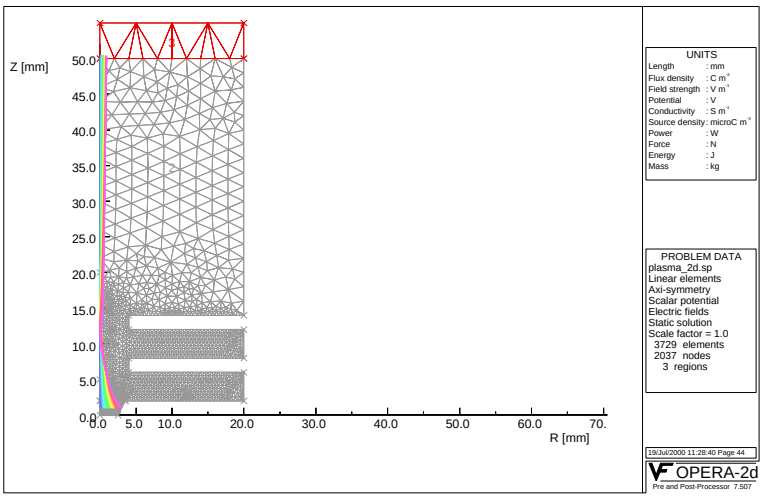


Figure 6.28 2d model of a simple rotational structure

We assume that the plasma mostly exists below Z=0 and that the emission surface will be somewhere in the cone-shaped opening in the electrode above the plasma ($0 < Z < 2$), see Figure 6.29. Note that the shape code of region 1 is set to Q, which will give a regular mesh.

The emitter is specified by giving a curve (in this case, a straight line) that will be anchored at its first point but is free to move until a self-consistent solution is obtained. By self-consistent, what we actually mean is that the normal electric field to the emitter surface remains constant along the surface. Figure 6.30 shows the contours of EMOD and the trajectories, where it is seen that although the surface has a constant normal electric field, it is not at a constant value of EMOD. Figure 6.31 shows contours of EZ and the trajectories.

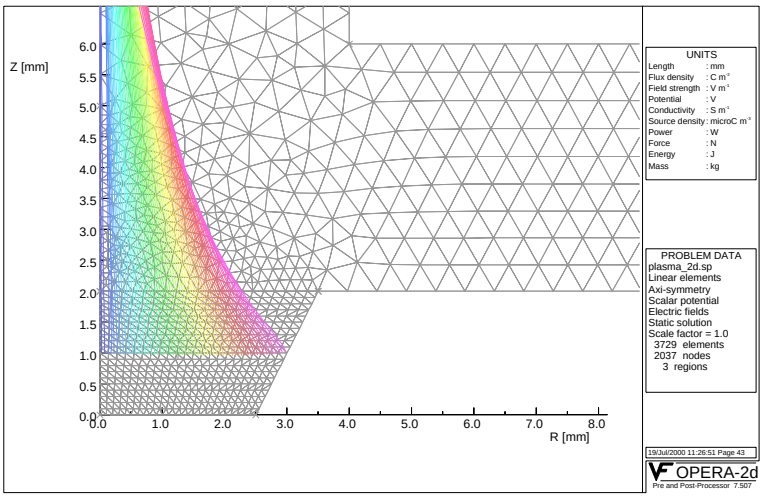


Figure 6.29 Emission surface

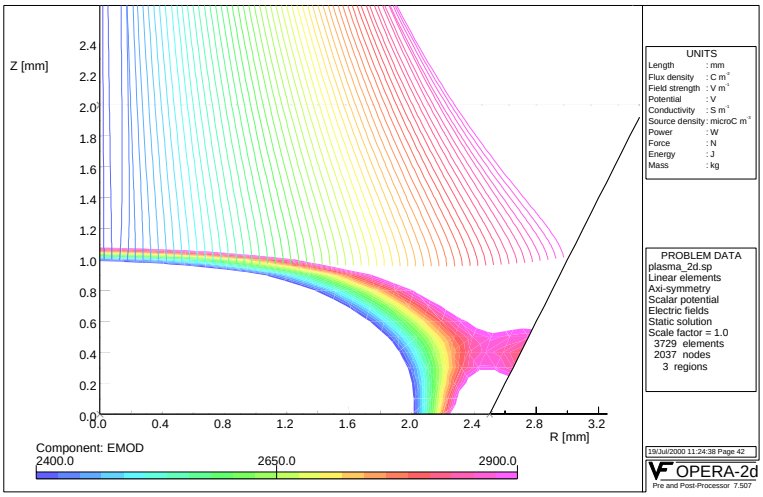


Figure 6.30 Contours of EMOD and emission surface

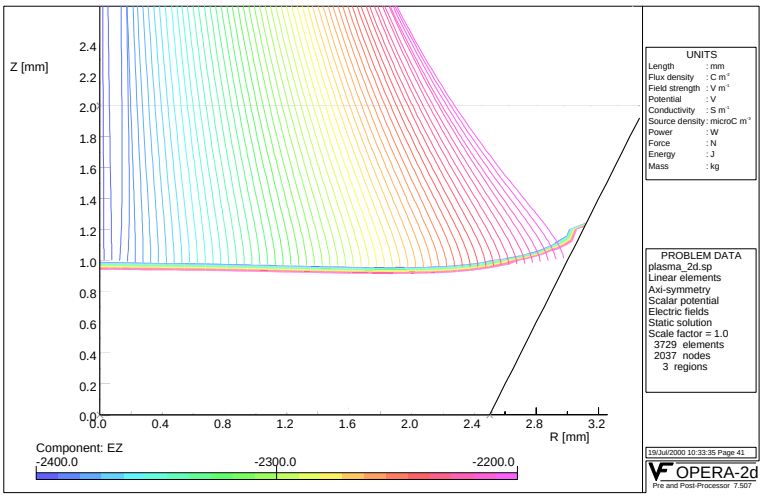


Figure 6.31 Contours of EZ and emission surface

Radial component of velocity

The particles are initially starting from a straight line, and all particles initially have only a z-component of the velocity (the x or r-component of the velocity being zero). At the end of the iteration process the emitter will have a curved surface, and there will be a velocity in the radial direction. Figure 6.32 shows the relative radial component (**VELX/#VELMOD**) of the velocity. Mostly the radial component is about zero (which is why it appears to be emitted from a constant EZ contour) but at the outer edge the radial velocity at the start of the track is negative.

The absolute value of the velocity has been specified with the expression

```
$PARA NAME=#velmod,VALUE=sqrt(velx**2+vely**2+velz**2).
```

The files required to experiment with this example are on the distribution CD: *plasma_2d.op2* and *plasma_2d.emit.*

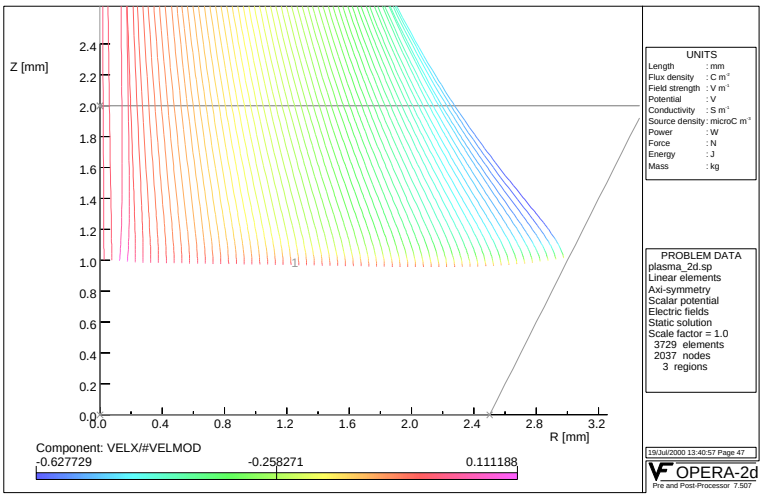


Figure 6.32 Relative radial component of the velocity

Chapter 7
A Transformer Example

Introduction

The transformer example models a step-down transformer. It has a primary and secondary winding and an E-I shape core. Since OPERA-2d models a two dimensional cross-section of an object, the transformer is assumed to be infinitely long in the 3rd dimension i.e. no end effects may be calculated. This is a reasonable assumption for this type of device, provided the yoke does not saturate.

The OPERA-2d pre and post processor provides a variety of sophisticated tools to simplify the modelling process. Some of these will be introduced in this example. In this case, the complete cross-section will be modelled using simple methods of construction. Seven regions will be used to create the model as shown in Figure 7.1.

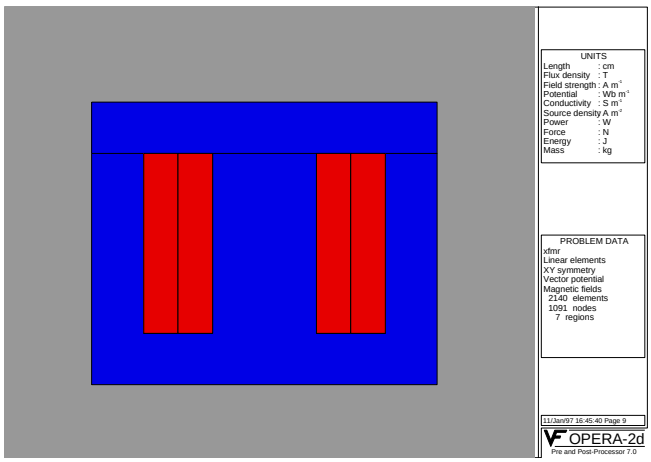


Figure 7.1 The complete model

Pre Processing - Building the Model

Entering the Pre and Post Processor

Start the pre and post processor in the usual way.

When the pre and post processor starts a default set of (SI) units is available. The user can change the default units to a more appropriate set of units if required.

The most convenient length units for a small transformer are centimetres. The length units can be changed from metres to centimetres as follows:-

click on

UNITS ↓
Length unit → **Centimetre**
Length unit → **Return**
Return

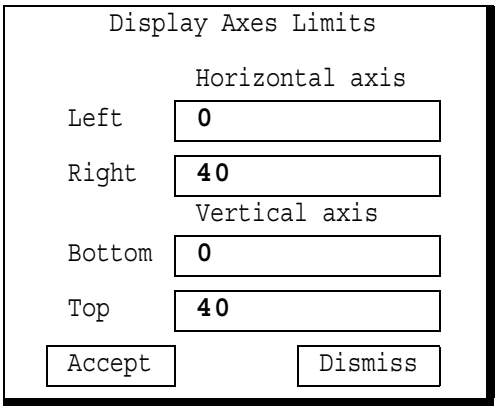
The scale displayed on the axes will still be shown as being in metres instead of centimetres. This will be changed to the current set of units, i.e. centimetres, next time the axes are redrawn.

There is also a default axis size of 10 units high (Y) and 10 units wide (X), with a 1:1 aspect ratio. The X axis may be longer than 10 units due to the aspect ratio of the graphics window. The display area may be changed to allow the entire transformer to be displayed[1]. To do this click on

DISPLAY ↓
Axes limits

and complete the box as shown below.

1. The new axis sizes shown here also provide a 1:1 aspect ratio. However, any other aspect ratio may be specified



Select the **Accept** button when it is complete, followed by:

DISPLAY ↓
Refresh

The graphics display will be redrawn with the correct set of units and axis dimensions. The size of the axes may be changed at any time without affecting the model (other than the way it is displayed in the graphics window). The units can also be changed at any time, without affecting the true size of the model.

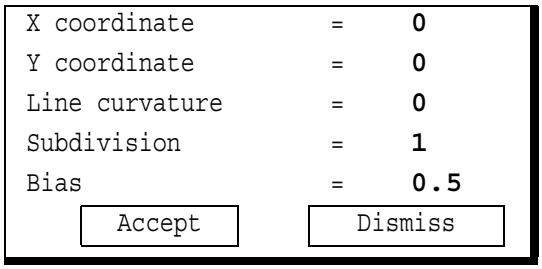
The units are now set correctly and construction of the model can begin. Each part is made up of one or more regions.

Drawing the Model Geometry

The first region to be drawn is the “E” part of the core. Since there were no existing points, an initial x, y coordinate of (0,0) will be used followed by relative moves in the x and y directions. The material characteristic will be changed from air (the default) to steel later in the tutorial, using the region modification. To begin drawing the region select

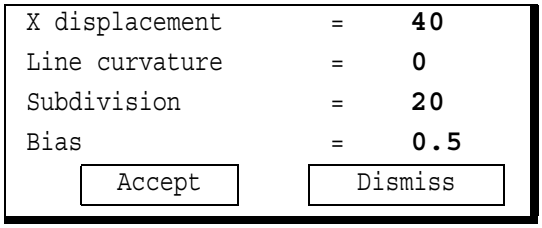
MODEL ↓
Draw regions → **New region ...**
... Polygon → **XY input**

and complete the dialog box as shown below.

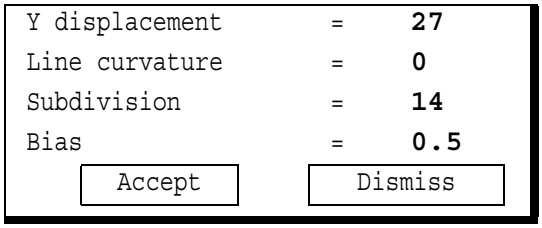


When complete, select **Accept**. Continue with the polygon definition by selecting the x and y moves as follows

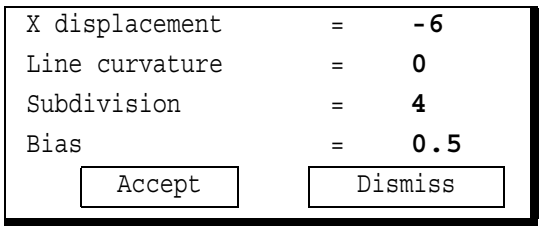
Draw regions → New region ...
... Polygon → X Move



Draw regions → New region ...
... Polygon → Y Move



Draw regions → New region ...
... Polygon → X Move



Draw regions → New region ...
... Polygon → Y Move

Y displacement	=	**-21**
Line curvature	=	**0**
Subdivision	=	**11**
Bias	=	**0.5**

Accept Dismiss

Draw regions → New region ...
... Polygon → X Move

X displacement	=	**-4**
Line curvature	=	**0**
Subdivision	=	**4**
Bias	=	**0.5**

Accept Dismiss

Draw regions → New region ...
... Polygon → X Move

X displacement	=	**-4**
Line curvature	=	**0**
Subdivision	=	**4**
Bias	=	**0.5**

Accept Dismiss

Draw regions → New region ...
... Polygon → Y Move

Y displacement	=	**21**
Line curvature	=	**0**
Subdivision	=	**11**
Bias	=	**0.5**

Accept Dismiss

Draw regions → New region ...
... Polygon → X Move

X displacement	=	**-12**
Line curvature	=	**0**
Subdivision	=	**6**
Bias	=	**0.5**
Accept		Dismiss

Draw regions → New region ...
... Polygon → Y Move

Y displacement	=	**-21**
Line curvature	=	**0**
Subdivision	=	**11**
Bias	=	**0.5**
Accept		Dismiss

Draw regions → New region ...
... Polygon → X Move

X displacement	=	**-4**
Line curvature	=	**0**
Subdivision	=	**4**
Bias	=	**0.5**
Accept		Dismiss

Draw regions → New region ...
... Polygon → X Move

X displacement	=	**-4**
Line curvature	=	**0**
Subdivision	=	**4**
Bias	=	**0.5**
Accept		Dismiss

Draw regions → New region ...
... Polygon → Y Move

Y displacement	=	**21**
Line curvature	=	**0**
Subdivision	=	**11**
Bias	=	**0.5**

Accept Dismiss

Draw regions → New region ...
... Polygon → X Move

X displacement	=	**-6**
Line curvature	=	**0**
Subdivision	=	**4**
Bias	=	**0.5**

Accept Dismiss

Draw regions → New region ...
... Polygon → Close Polygon

Subdivision	=	**14**
Line curvature	=	**0**
Bias	=	**0.5**

Accept Dismiss

Region 1 is now complete and is shown in Figure 7.2.

The “I” part of the core will be drawn as Region 2. Two points are input as XY coordinates and then use is made of existing points and X moves to complete the region. Region 2 is drawn as follows:-:

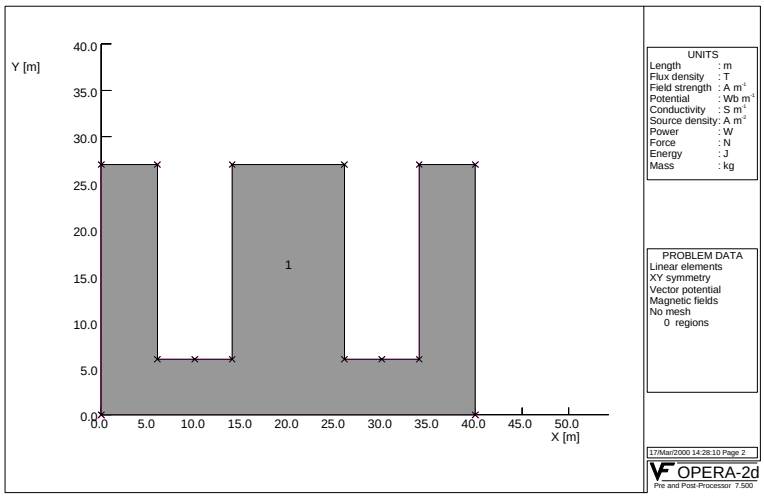


Figure 7.2 Region 1 - the E shaped core

MODEL ↓

Draw regions → **New region ...**

... Polygon → **XY input**

X coordinate	=	**0**
Y coordinate	=	**33**
Line curvature	=	**0**
Subdivision	=	**4**
Bias	=	**0.5**

Accept Dismiss

Draw regions → **New region ...**

... Polygon → **XY input**

X coordinate	=	**40**
Y coordinate	=	**33**
Line curvature	=	**0**
Subdivision	=	**20**
Bias	=	**0.5**

Accept Dismiss

Draw regions → **New region ...**

... Polygon → **Mouse input** → **At old point**

and position the mouse near the 2 coordinates shown below and press the mouse button at each.

40, 27
34, 27

To pick an existing point, it is only necessary to be nearer to the point to be selected than any other point. The cursor need not be directly on top of the old point.

In all future examples, the coordinates of points to be selected with the mouse will be shown as a coordinate pair, as above.

Draw regions → New region ...
... Polygon → Mouse input → Return

Select

Draw regions → New region ...
... Polygon → X Move

X displacement	=	**-4**
Line curvature	=	**0**
Subdivision	=	**4**
Bias	=	**0.5**
Accept		Dismiss

Draw regions → New region ...
... Polygon → Mouse input → At old point

and pick the points at the following positions:

26, 27
14, 27

Draw regions → New region ...
... Polygon → Mouse input → Return

Select

Draw regions → New region ...
... Polygon → X Move

X displacement	=	**-4**
Line curvature	=	**0**
Subdivision	=	**4**
Bias	=	**0.5**
Accept		Dismiss

Draw regions → New region ...
... Polygon → Mouse input → At old point

6, 27
0, 27

Draw regions → New region ...
... Polygon → Mouse input → Close polygon

Region 2 is completed. Regions 1 and 2 make up the whole of the core and this is shown in Figure 7.3.

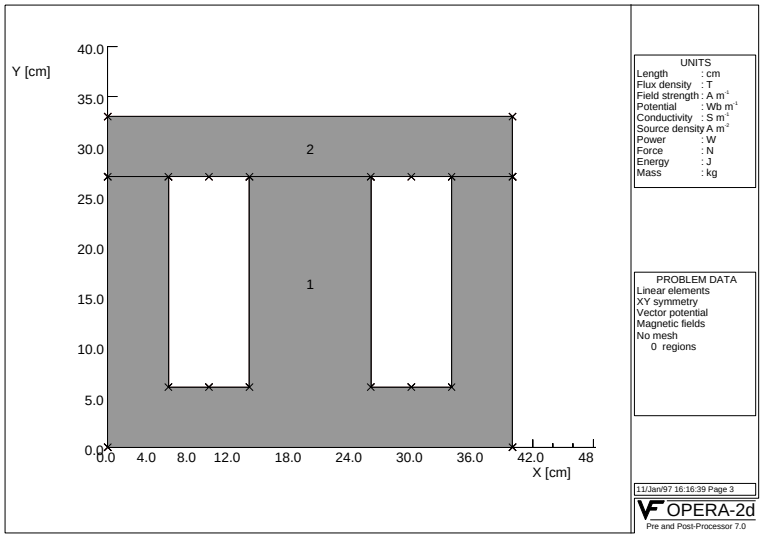


Figure 7.3 The Whole of the Core made up from Regions 1 and 2

The windings will be made up of four separate regions. Regions 3 and 6 will be the secondary windings and regions 4 and 5 will be the primary windings. Two regions are needed for each winding as one will carry the positive current and the other will carry the negative (return) current. The current density and material characteristics will be assigned after the regions are created. Existing points will be utilized in drawing the regions. To draw region 3 select

Draw regions → New region ...
... Polygon → Mouse input → At old point
6, 27
10, 27
10, 6
6, 6
Draw regions → New region ...
... Polygon → Mouse input → Close polygon

Region 3 created. Continuing:

```
Draw regions → New region ...
                    ... Polygon → Mouse input → At old point
10,    27
14,    27
14,    6
10,    6
Draw regions → New region ...
                    ... Polygon → Mouse input → Close polygon
```

Region 4 created. Continuing:

```
Draw regions → New region ...
                    ... Polygon → Mouse input → At old point
26,    27
30,    27
30,    6
26,    6
Draw regions → New region ...
                    ... Polygon → Mouse input → Close polygon
```

Region 5 created. Continuing:

```
Draw regions → New region ...
                    ... Polygon → Mouse input → At old point
30,    27
34,    27
34,    6
30,    6
Draw regions → New region ...
                    ... Polygon → Mouse input → Close polygon
```

Region 6 created. This completes the construction of the transformer geometry as shown in Figure 7.4.

Return to the MODEL menu by:

```
Draw regions → New region ...
                    ... Polygon → Mouse input → Return
Draw regions → New region ...
                    ... Polygon → Return
Draw regions → Return
```

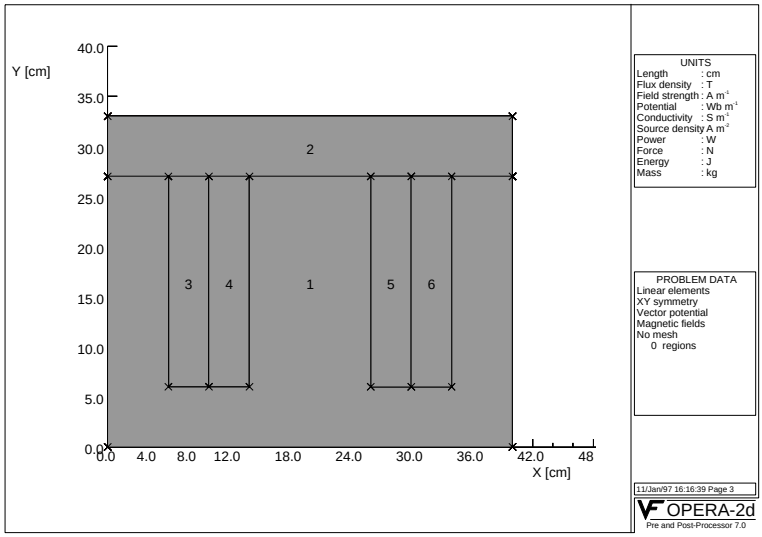


Figure 7.4 Regions 1 to 6 showing the Core and Windings

Modifying Regions to Include Material Data

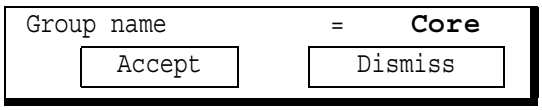
Although the geometry is complete, all the regions have material properties set to default values (corresponding to air)[1]. These need to be changed to correctly model the electromagnetic characteristics of the transformer. One hundred materials are available, Material label 0 represents air, Material label 1 represents non-magnetic conductors and Material labels 3 through 102 represent magnetic materials. (Material 2 is not used).

Once a region has been created, any changes to that region may be made with the **`Modify regions`** submenu of the **`MODEL`** menu. To assist in the modifications to the core, the 2 regions that have been created to model this will be grouped together.

Select

`MODEL` ↓

`Group regions` → **`Create a new group`**



`By cursor...`

`... add/remove regions` → **`Pick region to add`**

1. These default properties are Material label 0, mu or epsilon = 1, Density = 0, Conductivity = 0, Phase/angle = 0 and Velocity = 0.

and select the 2 regions of the core by

20, 15
20, 30

Each of the regions is highlighted as it is selected. Close the menu by

By cursor...
... add/remove regions → Return
Group regions → Create a new group → Return
Group regions → Return

The material properties of these 2 regions can then be changed together by selecting

MODEL ↓
Modify regions → Region group → CORE → Material data

Material label	=	**3**
Mu or epsilon	=	**1000**
Density	=	**0**
Conductivity	=	**0**
Phase/angle	=	**0**
Velocity	=	**0**

Accept Dismiss

After **Accept** and **Return**, the two regions that make up the group, CORE, are highlighted in blue.

As each of the conductors will have a different current density, it would not be advantageous to group them. Each region will be changed individually to have the correct current density and material label by

Select

Modify regions → Region group → Return
Modify regions → Modify Region → Pick Region

and then select region 3 by *clicking* the mouse anywhere inside region 3.

Modify regions →Modify region → Pick region → Material data

Material label	=	**1**
Mu or epsilon	=	**1**
Density	=	**1e8/area**
Conductivity	=	**0**
Phase/angle	=	**0**
Velocity	=	**0**

Accept Dismiss

Modify regions →Modify region → Pick region → Return

Region 3 turns red.

MODEL ↓

Modify regions → Modify Region → Pick Region

and then select region 4 by *clicking* the mouse anywhere inside region 4.

Modify regions →Modify Region → Pick Region → Material data

Material label	=	**1**
Mu or epsilon	=	**1**
Density	=	**-9.5e7/area**
Conductivity	=	**0**
Phase/angle	=	**0**
Velocity	=	**0**

Accept Dismiss

Modify regions →Modify region → Pick region → Return

region 4 is highlighted red.

MODEL ↓

Modify regions → Modify Region → Pick Region

and then select region 5 by *clicking* the mouse anywhere inside region 5.

Modify regions →Modify Region → Pick Region → Material data

Material label	=	**1**
Mu or epsilon	=	**1**
Density	=	**9.5e7/area**
Conductivity	=	**0**
Phase/angle	=	**0**
Velocity	=	**0**

Accept Dismiss

Modify regions →Modify Region → Pick Region → Return

Region 5 is highlighted red.

MODEL ↓

Modify regions → Modify Region → Pick Region

and then select region 6 by *clicking* the mouse anywhere inside region 6.

Modify regions →Modify Region → Pick Region → Material data

Material label	=	**1**
Mu or epsilon	=	**1**
Density	=	**-1e8/area**
Conductivity	=	**0**
Phase/angle	=	**0**
Velocity	=	**0**

Accept Dismiss

Modify regions →Modify Region → Pick Region → Return

Region 6 turns red.

Return to the top level menu by selecting

Modify regions → Modify Region → Return
Modify regions → Return
Return

The current densities in regions 3 and 4 are in opposite direction to the current in regions 5 and 6. This represents the return path of the winding.

The transformer has been modelled with the correct material types as shown in Figure 7.5.

Magnetic materials (labelled 3 through 102) must have a BH table associated with

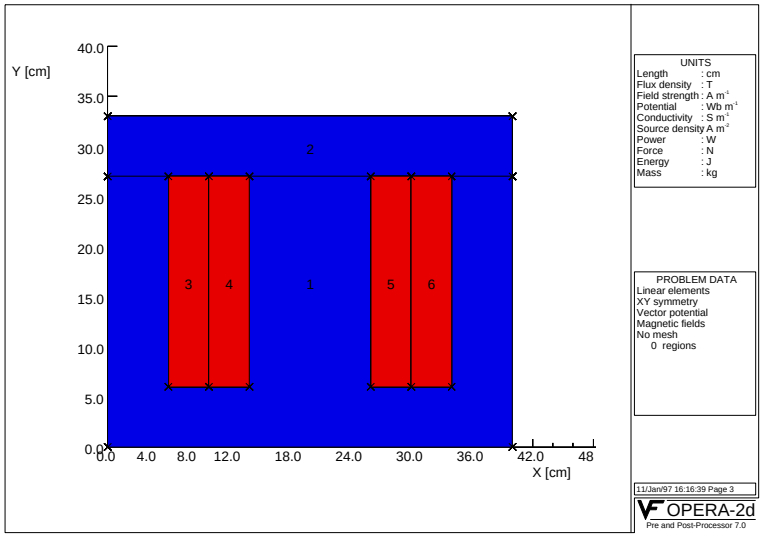


Figure 7.5 The Model including Material Specification

each material. Material label 3 (used here) has a default BH table (*default.bh*) associated with it. If any other magnetic material was used, a BH table would have to be assigned to the material label.[1]

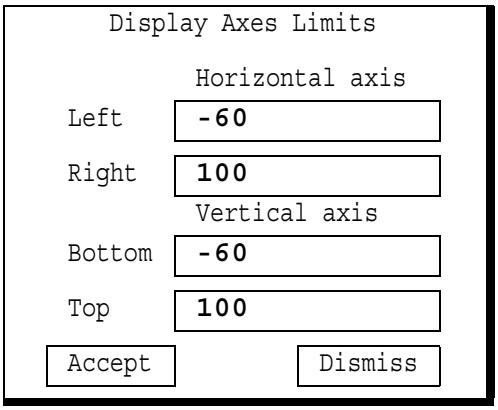
Only the surrounding air is left to be modelled. In finite element models, it is usually necessary to model air surrounding the electromagnetic device so that stray fields may be calculated. The air will be modelled using a single Background region. This is modelled so that it sits on top of regions 1 to 6, and extends to a limit where the stray field will be negligible.

The air region is larger than the current display size, so the screen coordinates should be changed:

```
DISPLAY ↓
      Axes limits
```

1. OPERA-2d is supplied with a library of BH characteristics. These are found in the $vfdir/bh directory of your computer system ($vfdir is the directory containing OPERA-2d. See “Implementation Notes” .)

Complete the box as shown below.



DISPLAY ↓

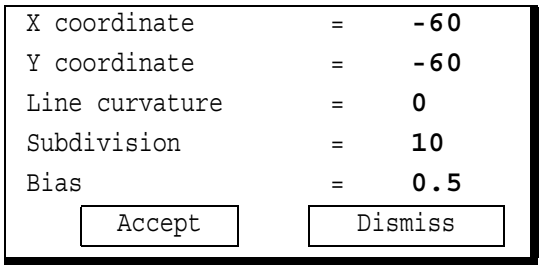
Refresh

To draw the air surrounding the transformer:

MODEL ↓

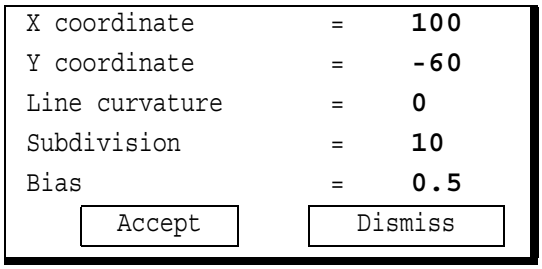
Draw regions → **New region ...**

... Background → **XY input**



Draw regions → **New region ...**

... Background → **XY input**



Draw regions → New region ...
... Background → XY input

X coordinate	=	**100**
Y coordinate	=	**100**
Line curvature	=	**0**
Subdivision	=	**10**
Bias	=	**0.5**
Accept		Dismiss

Draw regions → New region ...
... Background → XY input

X coordinate	=	**-60**
Y coordinate	=	**100**
Line curvature	=	**0**
Subdivision	=	**10**
Bias	=	**0.5**
Accept		Dismiss

Draw regions → New region ...
... Background → Close Polygon

Subdivision	=	**10**
Line curvature	=	**0**
Bias	=	**0.5**
Accept		Dismiss

Region 7 has been created. Only one background region may be used in a model. Hit any key to clear this message.

Select

Draw regions → New region ...
... Background → Return
Draw regions → Return

to return to the **MODEL** menu

The geometry, including material data and surrounding air is shown in Figure 7.6.

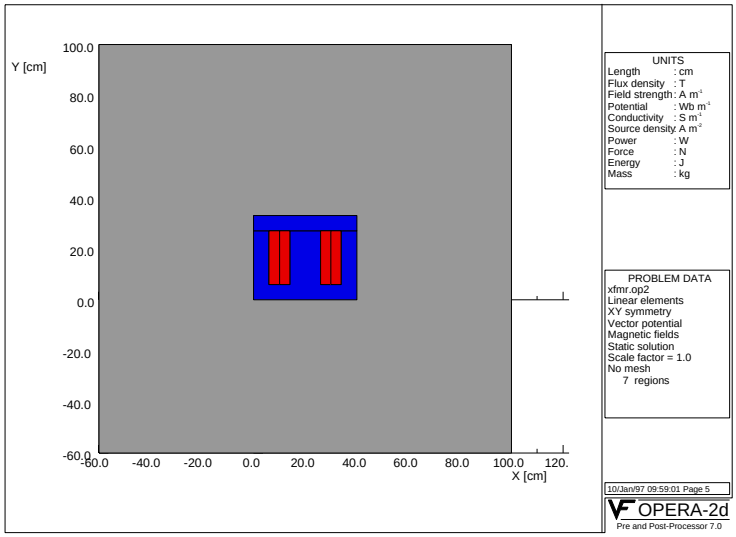


Figure 7.6 The model with the surrounding air

Setting the Boundary conditions

Following material data definition, the boundary conditions should be assigned. The boundary conditions are used in this case to approximate the magnetic field at large distances from the problem (far-field boundaries). For vector potential solutions it is generally necessary to assign at least one exterior region face as **`B normal = 0`**[1]. In this case, all exterior faces will be set by

`MODEL` ↓

`Boundary cond.` → **`Vector pot`** → **`B normal=0`**

and click around the 4 sides of the background region, inside the region.

20, 95
95, 20
-55, 20
20, -55

and then return to the **`MODEL`** menu by

`Boundary cond.` → **`Vector pot`** → **`Return`**
`Boundary cond.` → **`Return`**

1. Using **`Bnormal = 0`** boundary condition sets the vector potential along that boundary to be zero. This gives a unique value of vector potential to the problem, i.e. it *gauges* the solution.

Generating the Finite Element Mesh

The model with assigned boundary conditions has now been created. The next step is to generate and check the finite element mesh. The **Mesh generator** option of the **MODEL** menu generates the data necessary for the analysis modules to solve the problem. In addition, it checks the data and if any errors are found, the mesh will not be stored. To do this click on

MODEL ↓
Mesh generator → **Generate mesh**

A message box is created while the mesh is being generated. This displays the results of the checking procedure. If anything other than 0 errors are reported, these errors will need to be corrected. If anything other than 0 warnings are reported, these may need to be corrected.

The window displays an outline of the model. The boundary of the model is drawn with a red line. Red lines that appear inside the outer boundary indicate that the mesh is not continuous (i.e. an internal boundary has been found)[1].

Clear the message box and return to the top level menu by clicking on

MODEL ↓
Return

Preparing for Analysis

The problem to be modelled has a constant drive current and hence can be solved using the static analysis module. From the main menu select

FILE ↓
Write file → **Analysis data** → **Static analysis**

A menu listing all static analysis options appears. Accept all defaults by hitting **Return** twice

Storing a problem

It is essential to store the model[2]. To save the model select

1. In electrostatic problems, internal boundaries may be required to represent e.g. electrodes.
2. This can be done at any time during the pre processing stage, and should be used regularly to back up your work while building the model

FILE ↓
Write file → **Write model**

In the file selection box enter **xfmr** and press the **Accept** button. This creates two OPERA-2d files containing the model and mesh data. These are *xfmr.op2* and *xfmr.mesh* respectively. The *.op2* file is required as input to the analysis program. The *.mesh* file is not required by the analysis program (although if it is present, it will be used by the analysis and thus speed up the analysis), so can be deleted to save disk space if desired. However if no mesh file exists then the **Remesh** option of the **FILE** menu must be used on entering the pre and post processor and before examining results.

To exit the pre and post processor:

FILE ↓
End OPERA-2d/PP

OPERA-2d Static Analysis Module

A number of analysis modules are supplied with OPERA-2d. The choice of analysis module depends on the type of problem being solved. (For further information see “Analysis and Utility Programs” on page 5-1). In this particular case, an instant in time of the transformer operation is considered and eddy currents are neglected, i.e. the core is assumed to be laminated. Hence it can be considered as a time invariant (static) magnetic field problem and the static analysis module is to be used.

In order to launch the Static Analysis module select

FILE ↓
Start analysis

Within the **Start analysis** menu select the appropriate filename **(xfmr)** and **Accept**.

OPERA-2d then reports on the progress of the solution.

On successful completion, OPERA-2d returns to the initial screen. The solution process has created a solution file, *xfmr.st*. This contains the solution data from the static analysis for post processing.

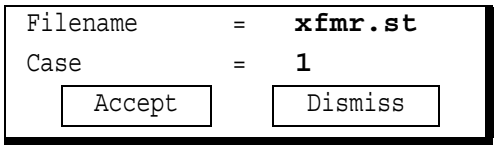
Post Processing - Examining the Solution

Reading and displaying the solution

To analyse the solution from the static analysis, the results file, *xfmr.st* is read into the pre and post processor. To do this click on

FILE ↓
Read file → **Read model**

A file box listing the directory contents appears, from which the file *xfmr.st* should be selected. The **Accept** button should then be pressed.



The file *xfmr.st* is read into the pre and post processor. A message box is displayed showing the units setting. This can be cleared by hitting any key or mouse button.

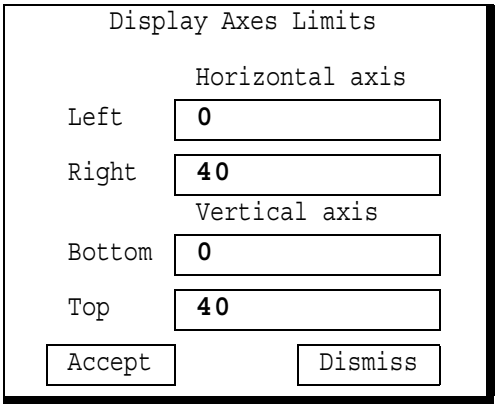
Normally the mesh file will have been created by the ST analysis, and should still be available. If the mesh file does not exist (for example if it has been deleted), it will be necessary to rebuild the mesh. To do this select

FILE ↓
Read file → **Re-mesh**

The remeshing process produces a message box. This may be cleared using any key.

The solution will not appear on the screen until the screen is redrawn. Using the **DISPLAY** menu, resize and redraw the screen:

DISPLAY ↓
Axes limits



DISPLAY ↓
Style → **Fill materials**

DISPLAY ↓
Style → **Return**

DISPLAY ↓
+Labels *(toggles to -Labels)*

DISPLAY ↓
Nodes → **No vertices/nodes**

DISPLAY ↓
Nodes → **Return**

DISPLAY ↓
Refresh

The air regions overdraw the axes. To see the axes after the air regions have been drawn, choose the **Axes - Overdraw** options from the **DISPLAY** menu and then **Refresh**.

Line Contours of Vector Potential

One of the easiest things to check is the flux pattern. The static analysis is solved in terms of vector potential and equipotential line contours, in XY symmetry[1], are equivalent to lines of magnetic flux. To see contours of constant vector potential select

FIELDS ↓

Contour plot → **Execute**

A message box displaying the model RMS error is given. This can be cleared by hitting any key or mouse button.

The error can be significantly reduced by changing to quadratic elements in the pre processing stage

Figure 7.7 shows the lines of constant vector potential over the whole model.

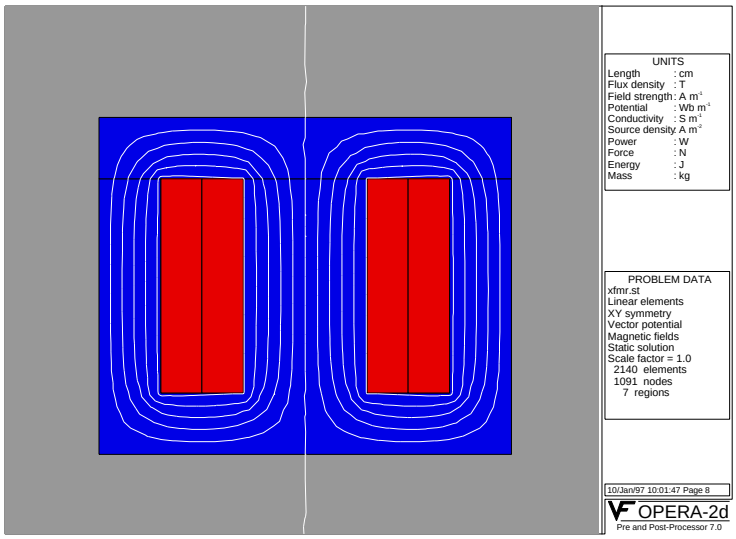


Figure 7.7 Equipotential line contours of vector potential - Equivalent to lines of magnetic flux

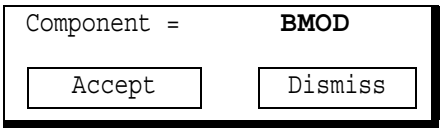
Zone Contours of Flux Density Magnitude

In addition to the magnetic lines of flux, the magnetic flux density is a good indicator of magnetic performance. The contours may be shown in a number of different styles, for example filled zones. This displays zones or bands of colour on the model geometry. Each colour band represents a range of values.

To look at flux density one must change the component to be analysed from the default (POT) to BMOD (modulus i.e. magnitude of magnetic flux density **B**) before selecting the contour plot. Begin by:

1. This is not the case for an axi-symmetry model where the magnetic lines of flux are equivalent to the product of radius and potential, i.e. R*POT, or lines of POT if a modified rA potential was used.

Component



DISPLAY ↓
Contour plot → Style → Filled zones
DISPLAY ↓
Contour plot → Style → Return
DISPLAY ↓
Contour plot → No Refresh (*toggles to Refresh)*
DISPLAY ↓
Contour plot → Execute

The magnitude of magnetic flux density is shown in Figure 7.8

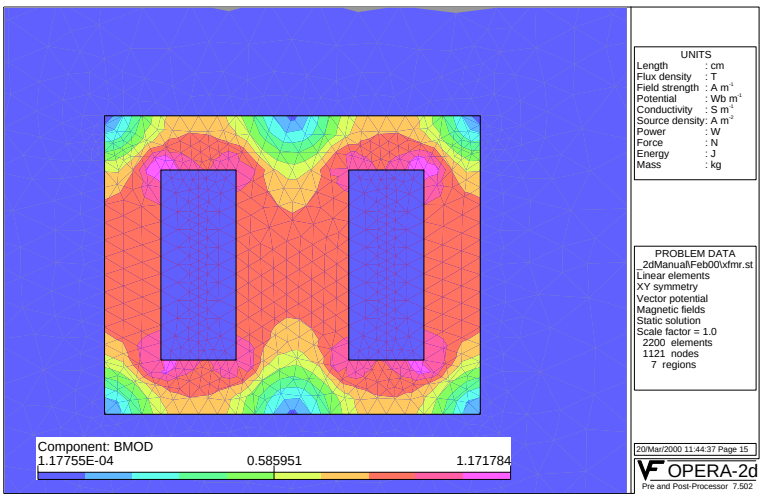


Figure 7.8 Filled Zone Contours of Magnetic Flux Density Magnitude

The display is redrawn and colour bands of flux density are drawn on the model and on the scale at the bottom of the graphics window.

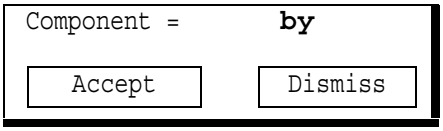
To just study the flux in the core select

FIELDS ↓
...Options (under Contour plot) → Region group → CORE
...Options (under Contour plot) → Return
Contour plot → Execute

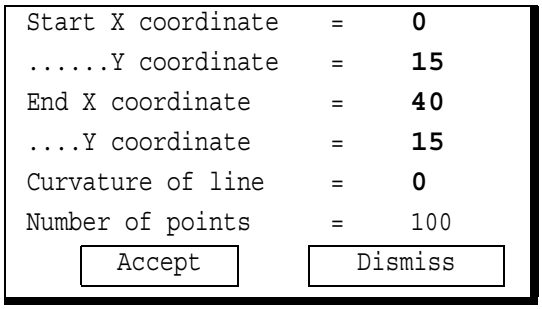
Line Graphs

Graphs representing lines in the two dimensional space may also be drawn. The appropriate component needs to be selected prior to drawing the graph. To see a line graph of the y-component of flux density through the middle of the transformer:

Component



Graphs → Along line



The resulting graph is shown in Figure 7.9.

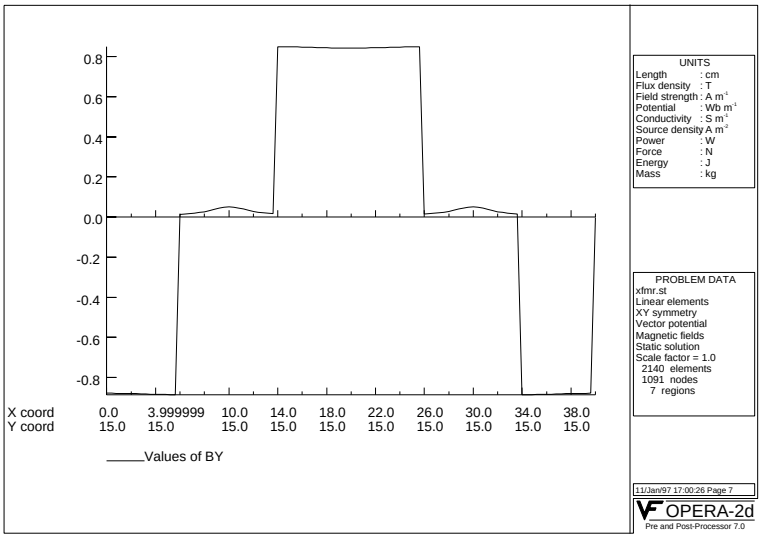


Figure 7.9 A Graph of the Y Component of Flux Density

This concludes the transformer example. To close the pre and post processor

```
FILE ↓
     End OPERA-2d/PP
```

Alternatively, move on to the next example. Before starting it is advisable to clear and reset the pre and post processor:

```
OPTIONS ↓
     Clear and reset
```

Chapter 8
A Solenoid Example

Introduction

The solenoid tutorial models a cylindrical solenoid which is rotationally symmetric about the vertical (z) axis. By making use of this symmetry only half (the positive R) of the 2D cross-section of the solenoid is modelled as shown in Figure 8.1,

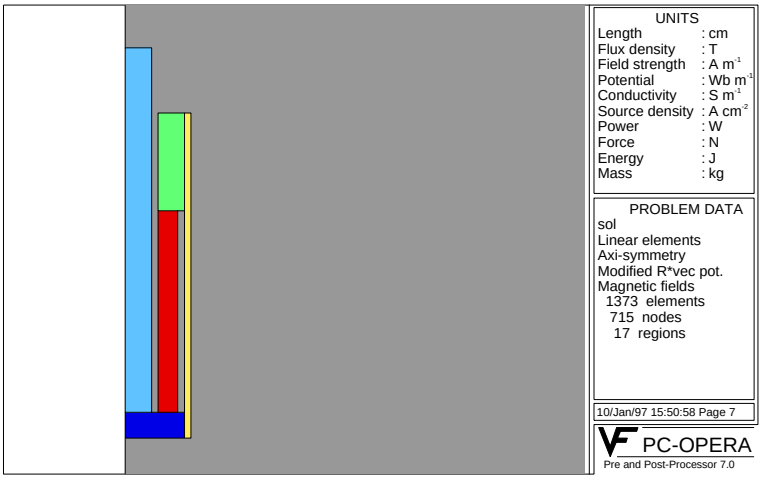


Figure 8.1 The Complete Model

The solenoid consists of four parts together with the coil. They are a base plate, a plunger, a top plate, and an outer shell.

Each part could be modelled as a single polygon. However the aspect ratio of 1:6 would lead to a poor quality finite element mesh. Therefore each part will made up of several quadrilateral regions which can each be discretized into a uniform and predictable mesh.

Each part will be given a different material label. The different labels usually signify different material characteristics, though in this case they are used as an aide in identifying the different parts of the model, as each has a different colour associated with it. (All the magnetic materials in this model have the same BH characteristics.)

OPERA-2d Pre Processor

Setting the Working Environment

Launch OPERA-2d and enter the pre and post processor as described in previous examples.

It is necessary to change the default set of units, select the axis dimensions and set the solution type. The solenoid will be drawn in centimetres with the current density in Acm^{-2}. To change the default system of units:

UNITS ↓
Length unit → Centimetre
Length unit → Return
Density unit → Amps/cm2**
Density unit → Return
Return

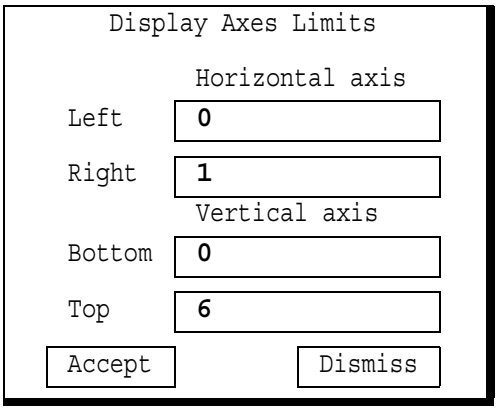
and the units are changed.

The default screen coordinates should be changed. Since the solenoid is essentially a long cylinder, an aspect ratio of 1:6 will assist in viewing the model being constructed.

Note: This aspect ratio will distort the view of the model but will allow the user to see more detail.

Select

DISPLAY ↓
Axes limits



and select the **Accept** button to set the new size, followed by **Refresh** to update the display.

The default solution type is XY SYMMETRY. Since the model has axi-symmetry, the solution type needs to be changed to axi-symmetry with solution potential MODIFIED RA.

With axi-symmetry, the modified rA potential is used for greater accuracy around the Z-axis. Select

MODEL ↓
Solution Type → Axi symm and potentials → Modified r*A
Solution Type → Axi symm and potentials → Return
Solution type → Return

The screen display can be refreshed to show these changes by selecting

MODEL ↓
Zoom display → Same size

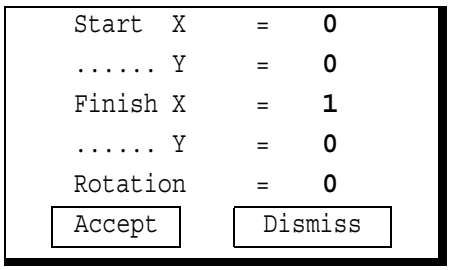
The user should always check that the symmetry and solution are set correctly before continuing. This can be done by checking the text information written on the right hand side of the display.

Construction Lines Entry

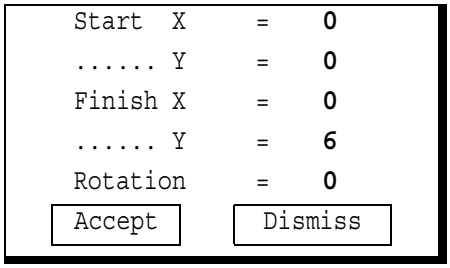
Construction lines will be used to aid in drawing regions. Several of these lines are created by selecting

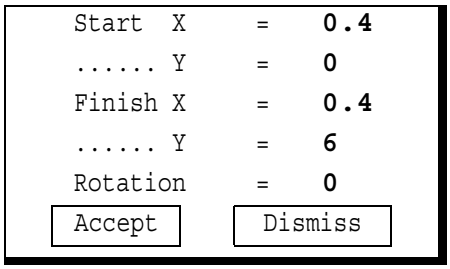
MODEL ↓
Draw regions → New region ...
... Polygon → Enter C_lines → Line

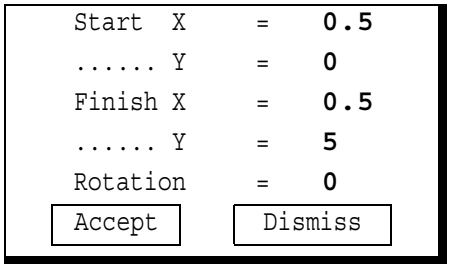
and completing the box as shown below.

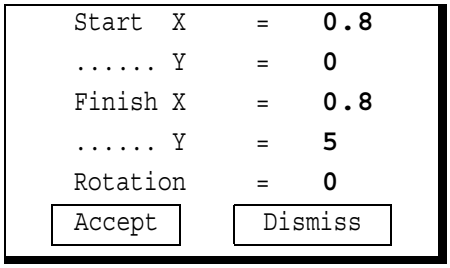


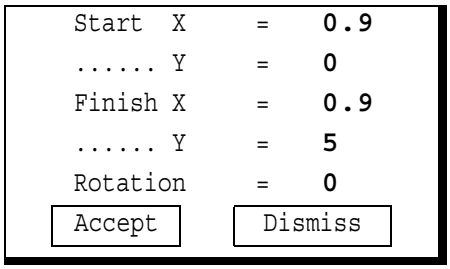
When complete, select **Accept**. The construction line is drawn from (0,0) to (1,0). Repeat for the following construction lines.

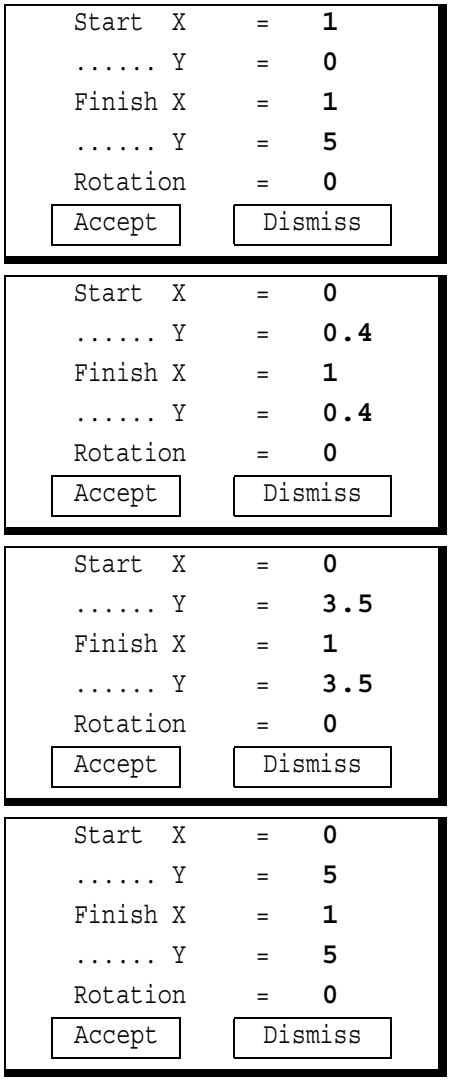












The construction lines will not be a part of the model, and will not be stored in the *.op2* file. They are an aid to drawing the regions.

Select

Draw regions → New region ...
... Polygon → Enter C_lines → Return
Draw regions → New region ...
... Polygon → Return

Drawing the Regions

The model is constructed by placing region vertices at the intersections and ends of the construction lines (the default region type is the polygon which can have up to 1000 sides).

Four regions will be drawn to represent the base plate of the solenoid, as illustrated in Figure 8.2. The material properties associated with these regions should

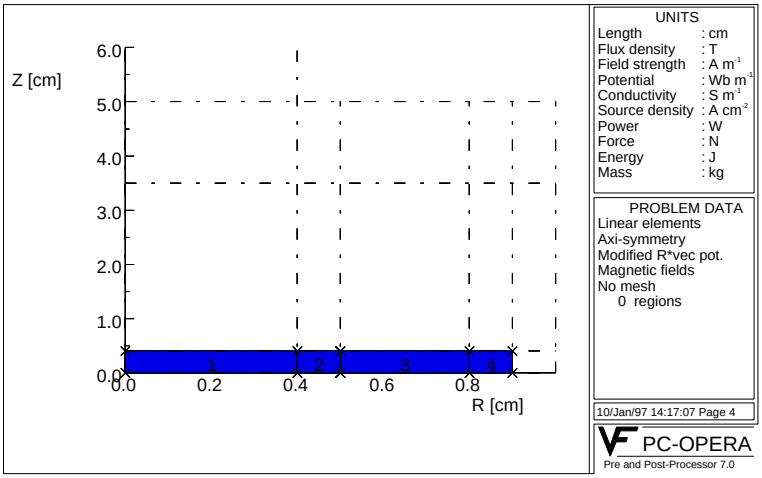


Figure 8.2 The base plate geometry

be set before the regions are drawn. In this case, the base plate will have a material label of 3 and an initial permeability of 1000. This is done by

MODEL ↓

Draw regions → Region defaults ...

... material type

Material label	=	3
Mu or epsilon	=	1000
Density	=	0
Conductivity	=	0
Phase/angle	=	0
Velocity	=	0

Accept Dismiss

The polygons are now drawn by selecting

Draw regions → New region ...
... Polygon → Mouse input → At intersection

and then clicking the mouse near the following coordinates, near the intersections of the construction lines[1]

0, 0
0.4, 0
0.4, 0.4
0, 0.4

Draw regions → New region ...
... Polygon → Mouse input → Close polygon

Region 1 is created.

Draw regions → New region ...
... Polygon → Mouse input → At intersection

and then clicking the mouse near the following coordinates

0.4, 0
0.5, 0
0.5, 0.4
0.4, 0.4

Draw regions → New region ...
... Polygon → Mouse input → Close polygon

Region 2 is created.

Draw regions → New region ...
... Polygon → Mouse input → At intersection

and then clicking the mouse near the following coordinates

0.5, 0
0.8, 0
0.8, 0.4
0.5, 0.4

Draw regions → New region ...
... Polygon → Mouse input → Close polygon

Region 3 is created.

Draw regions → New region ...
... Polygon → Mouse input → At intersection

1. *At intersection* will create a point where 2 construction lines meet or at the end of a construction line,

and then clicking the mouse near the following coordinates

0.8, 0
0.9, 0
0.9, 0.4
0.8, 0.4

Draw regions → New region ...
... Polygon → Mouse input → Close polygon

Region 4 is created.

Draw regions → New region ...
... Polygon → Mouse input → Return
Draw regions → New region ...
... Polygon → Return

The plunger geometry is shown in Figure 8.3. For the 3 regions that make up the plunger, the material properties should be set with material label 5.

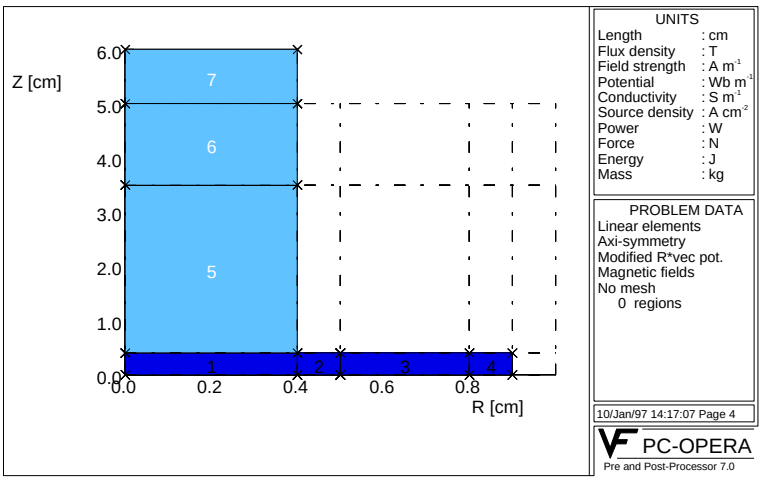


Figure 8.3 The first two components of the solenoid

Draw regions → Region defaults ...
... material type

Material label	=	**5**
Mu or epsilon	=	**1000**
Density	=	**0**
Conductivity	=	**0**
Phase/angle	=	**0**
Velocity	=	**0**

Accept Dismiss

Draw regions → New region ...
... Polygon → Mouse input → At intersection

and then clicking the mouse near the following coordinates

0, 0.4
0, 3.5
0.4, 3.5
0.4, 0.4

Draw regions → New region ...
... Polygon → Mouse input → Close polygon

Region 5 is created.

Draw regions → New region ...
... Polygon → Mouse input → At intersection

and then clicking the mouse near the following coordinates

0, 3.5
0, 5
0.4, 5
0.4, 3.5

Draw regions → New region ...
... Polygon → Mouse input → Close polygon

Region 6 is created.

Draw regions → New region ...
... Polygon → Mouse input → At intersection

and then clicking the mouse near the following coordinates

0, 5
0, 6
0.4, 6

0.4, 5

Draw regions → New region ...
... Polygon → Mouse input → Close polygon

Region 7 is created.

Draw regions → New region ...
... Polygon → Mouse input → Return
Draw regions → New region ...
... Polygon → Return

The top plate shown in Figure 8.4 should be assigned material label 7.

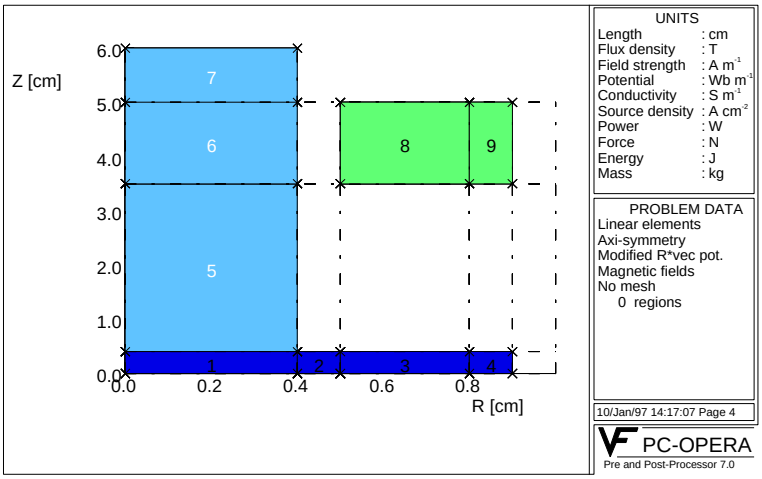


Figure 8.4 The modelling of the solenoid progresses

Draw regions → Region defaults ...
... material type

Material label	=	7
Mu or epsilon	=	1000
Density	=	0
Conductivity	=	0
Phase/angle	=	0
Velocity	=	0

Accept Dismiss

The 2 regions used to represent it are drawn by

```
Draw regions → New region ...
                  ... Polygon → Mouse input → At intersection
```

and then clicking the mouse near the following coordinates

0.5, 5
0.5, 3.5
0.8, 3.5
0.8, 5

```
Draw regions → New region ...
                  ... Polygon → Mouse input → Close polygon
```

Region 8 is created.

```
Draw regions → New region ...
                  ... Polygon → Mouse input → At intersection
```

and then clicking the mouse near the following coordinates

0.8, 5
0.8, 3.5
0.9, 3.5
0.9, 5

```
Draw regions → New region ...
                  ... Polygon → Mouse input → Close polygon
```

Region 9 is created.

```
Draw regions → New region ...
                  ... Polygon → Mouse input → Return
Draw regions → New region ...
                  ... Polygon → Return
```

Regions 10 to 12 will represent the outer shell of the solenoid. They will be labelled material 9. Continuing:

Draw regions → Region defaults ...
... material type

Material label	=	**9**
Mu or epsilon	=	**1000**
Density	=	**0**
Conductivity	=	**0**
Phase/angle	=	**0**
Velocity	=	**0**

Accept Dismiss

The menu boxes may cover up some of the coordinates required, e.g. (1,5). In order to point at them, the menu boxes must be hidden. Press the **F1** function key to hide the menus. Hitting the **F1** key again restores the menus. Use this feature when necessary in creating the following regions.[1]

Draw regions → New region ...
... Polygon → Mouse input → At intersection

and then clicking the mouse near the following coordinates

0.9, 5
0.9, 3.5
1, 3.5
1, 5

Draw regions → New region ...
... Polygon → Mouse input → Close polygon

Region 10 is created.

Draw regions → New region ...
... Polygon → Mouse input → At intersection

and then clicking the mouse near the following coordinates

0.9, 3.5
1, 3.5
1, 0.4
0.9, 0.4

Draw regions → New region ...
... Polygon → Mouse input → Close polygon

1. The F1 key acts as a *toggle* switch to hide and restore the menus

Region 11 is created.

Draw regions → New region ...
... Polygon → Mouse input → At intersection

and then clicking the mouse near the following coordinates

0.9, 0.4
0.9, 0
1, 0
1, 0.4

Draw regions → New region ...
... Polygon → Mouse input → Close polygon

Region 12 is created.

Draw regions → New region ...
... Polygon → Mouse input → Return
Draw regions → New region ...
... Polygon → Return

Region 13 will be the coil (conductor) region. Non-magnetic conductor regions are normally given material label 1. The current density flowing in the coil is set in the density parameter. Continuing:

Draw regions → Region defaults ...
... material type

Material label	=	**1**
Mu or epsilon	=	**1**
Density	=	**150**
Conductivity	=	**0**
Phase/angle	=	**0**
Velocity	=	**0**

Accept Dismiss

Draw regions → New region ...
... Polygon → Mouse input → At intersection

and then clicking the mouse near the following coordinates

0.8, 0.4
0.8, 3.5
0.5, 3.5
0.5, 0.4

Draw regions → New region ...
... Polygon → Mouse input → Close polygon

Region 13 is created.

```
Draw regions → New region ...
                    ... Polygon → Mouse input → Return
Draw regions →  New region ...
                    ... Polygon → Return
```

The regions drawn are illustrated in Figure 8.5

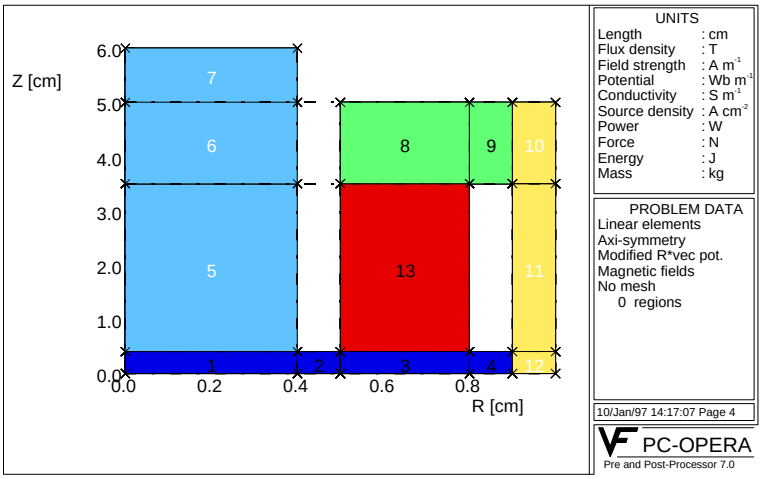


Figure 8.5 Regions 1 to 13 complete

Finally, only the surrounding air regions need to be drawn. This could be done using a single background region. However, the air regions inside the solenoid would be hard to mesh adequately, due once more to the high aspect ratio. Three quadrilateral regions will be used inside the model, and a single background region will extend to the far-field boundary.

First the material properties must be set to those of air. Air is normally given material label 0.

Draw regions → Region defaults ...
... material type

Material label	=	**0**
Mu or epsilon	=	**1**
Density	=	**0**
Conductivity	=	**0**
Phase/angle	=	**0**
Velocity	=	**0**

Accept Dismiss

To draw the 3 quadrilateral regions

Draw regions → New region ...
... Polygon → Mouse input → At intersection

and then clicking the mouse near the following coordinates

0.8, 0.4
0.8, 3.5
0.9, 3.5
0.9, 0.4

Draw regions → New region ...
... Polygon → Mouse input → Close polygon

Region 14 is created.

Draw regions → New region ...
... Polygon → Mouse input → At intersection

and then clicking the mouse near the following coordinates

0.5, 0.4
0.5, 3.5
0.4, 3.5
0.4, 0.4

Draw regions → New region ...
... Polygon → Mouse input → Close polygon

Region 15 is created.

Draw regions → New region ...
... Polygon → Mouse input → At intersection

and then clicking the mouse near the following coordinates

0.5, 3.5
0.5, 5
0.4, 5
0.4, 3.5

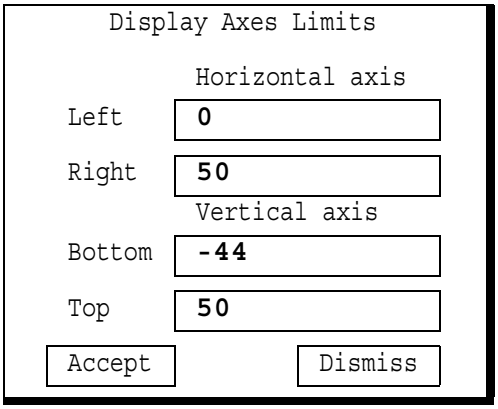
Draw regions → New region ...
... Polygon → Mouse input → Close polygon

Region 16 is created.

Draw regions → New region ...
... Polygon → Mouse input → Return
Draw regions → New region ...
... Polygon → Return
Draw regions → Return

The surrounding air region must be much larger than the solenoid so that stray fields can be accurately modelled. For this type of geometry a general rule requires the air to extend to 10 times the size of the model. The display should be resized by

MODEL ↓
Zoom display → Numerical Axes Limits



The background region is drawn by

MODEL ↓

Draw regions → New region ...

... Background → XY input

X coordinate	=	**0**
Y coordinate	=	**0**
Line curvature	=	**0**
Subdivision	=	**1**
Bias	=	**0.5**
Accept		Dismiss

Draw regions → New region ...

... Background → Y move

Y displacement	=	**6**
Line curvature	=	**0**
Subdivision	=	**10**
Bias	=	**0.5**
Accept		Dismiss

A bias is given to the next line to make the subdivision size smaller at the start of the face where the fields will be changing faster, and the elements need to be smaller.

Draw regions → New region ...

... Background → Y move

Y displacement	=	**44**
Line curvature	=	**0**
Subdivision	=	**10**
Bias	=	**0.01**
Accept		Dismiss

Draw regions → New region ...

... Background → XY input

X coordinate	=	**47**
Y coordinate	=	**3**
Line curvature	=	**1/47**
Subdivision	=	**5**
Bias	=	**0.5**
Accept		Dismiss

Draw regions → New region ...
... Background → XY input

X coordinate	=	**0**
Y coordinate	=	**-44**
Line curvature	=	**1/47**
Subdivision	=	**5**
Bias	=	**0.5**

Accept Dismiss

A bias will make the subdivisions smaller towards the end of the next face.

Draw regions → New region ...
... Background → Close polygon

Subdivision	=	**10**
Line curvature	=	**0**
Bias	=	**0.99**

Accept Dismiss

For mesh continuity within the model, the sides of neighbouring polygons must match exactly. In this case the background region just drawn has a side that does not match existing sides correctly. A message box appears listing the necessary points that should be added to make the neighbouring sides match, and asks if you would like these points to be added. Select **Yes**.

As only one background region may be used in a model, a further message appears, saying that any further regions drawn will be normal polygons. Hit any key to clear it.

Figure 8.6 shows the model with all regions entered.

Improving the Mesh

Return to the top level menu by

Draw regions → New region ...
... Background → Return
Draw regions → Return
Return

To display the finite element mesh in the parts of the solenoid

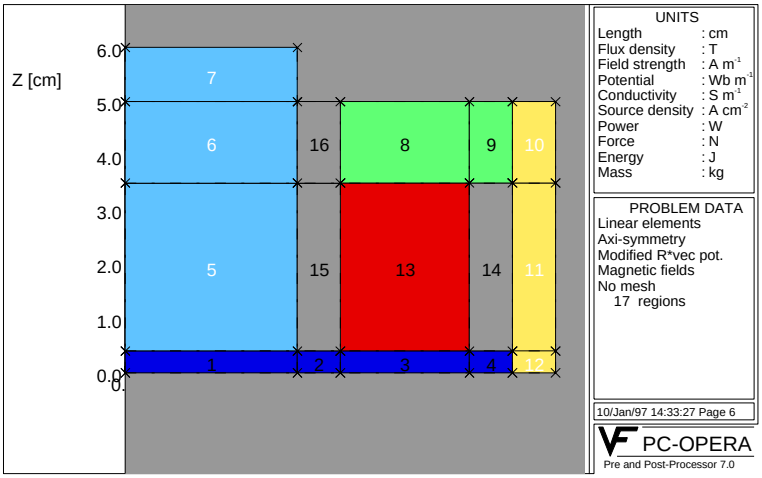
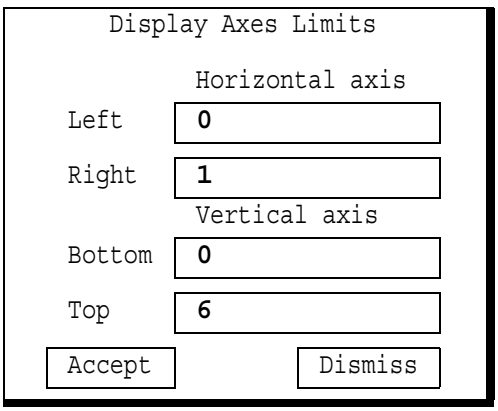


Figure 8.6 The completed model including surrounding air

DISPLAY ↓

-mesh ***(toggles to +mesh)***

Axes limits



DISPLAY ↓

Refresh

A message appears saying that the background mesh cannot be displayed. This is because the background region mesh has not yet been generated. Hit any key to clear this message.

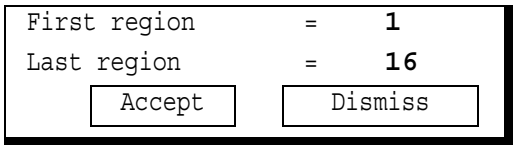
The mesh displayed is very sparse, many more elements are needed within the model. First the regions need to be changed so the mesh generator can produce a regular mesh.

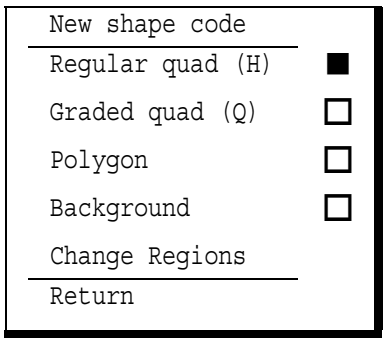
Other types of regions which can be created have codes H, Q, C and CS. These alternative region code definitions provide a method of generating a predictable mesh. In this example all the quadrilateral regions (1 to 16 inclusive) are to be changed from polygon regions. Most will be shape H. H shape regions produce a uniform mesh (opposite faces have an equal number of subdivisions).

To do this:

MODEL ↓

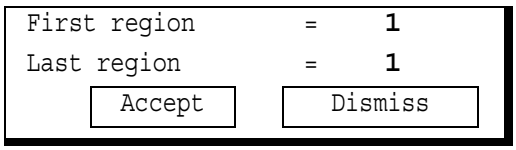
Change regions → **Region numbers**

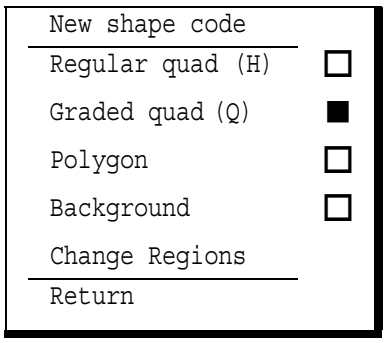




and select **Regular quad (H)** followed by **Change regions** to change regions 1 through 16 to shape H. Regions 1 and 5 will need many elements at their interface, but not as many opposite. They will be converted to shape Q regions. Q shaped regions allow graded mesh quadrilaterals. (One pair of opposite sides can have different number of subdivisions). To make this change:

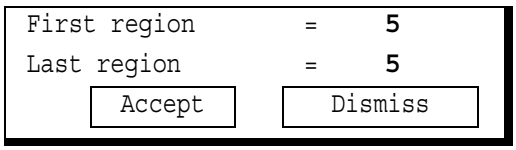
Change regions → **Region numbers**

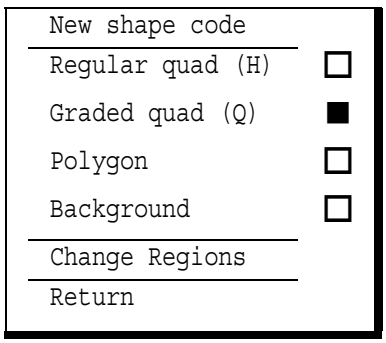




selecting **Graded quad (Q)** and **Change regions.**

Change regions → Region numbers





selecting **Graded quad (Q)** and **Change regions.**

The number of subdivisions are to be modified on the faces in each region. The number of subdivisions specifies the number of elements along each face. This information is used for mesh generation.

Select

DISPLAY ↓
Axes → Overdraw axes

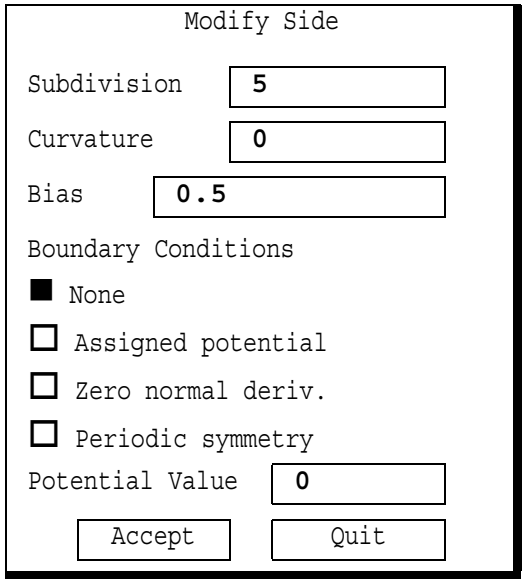
and **Refresh.**

MODEL ↓
Modify regions → Modify sides ...
... by picking → All properties...
...of one side

and select the side in region 1 near

0.2, 0

complete the parameter box:



MODEL ↓

Modify regions → **Modify sides ...**

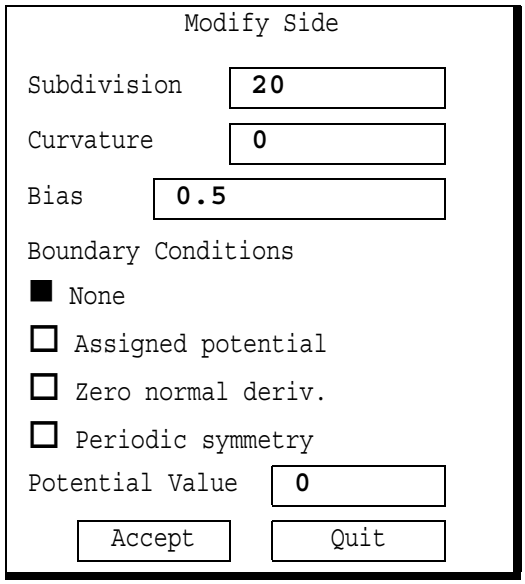
... by picking → **All properties...**

... of one side

Select another side in region 1 near

0.2, 0.4

and complete the parameter box:



Modify regions → **Modify sides ...**
... by picking → **Return**

To see the effect of this:

Modify regions → **Zoom display** → **Same size**

Continue improving the mesh using the **Modify sides** option used above, and change the subdivisions as shown in the table below.

Selection	Subdivision
Inside region 5 near (0.2,3.5)	5
Inside region 2 near (0.45,0.4)	3
Inside region 3 near (0.65,0.4)	5
Inside region 4 near (0.85,0.4)	2
Inside region 11 near (0.95,0.4)	3
Inside region 1 near (0,0.2)	5
Inside region 1 near (0.4,0.2)	5
Inside region 5 near (0,2)	8
Inside region 5 near (0.4,2)	8
Inside region 6 near (0.4,4.5)	8
Inside region 7 near (0.4,5.5)	4

and then return to the top level **COMMAND** menu.

To show the changes select

DISPLAY ↓
Refresh

Figure 8.7 shows the mesh that should exist after all the changes.

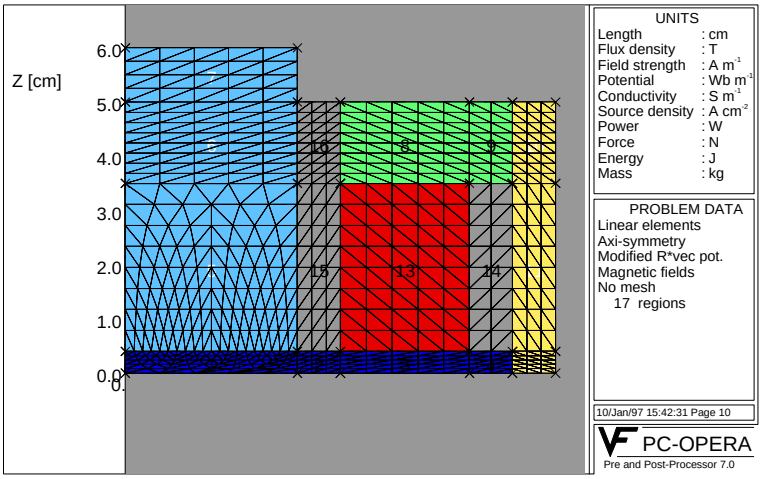


Figure 8.7 The mesh generated in the solenoid

Material Characteristics - the BH Curve

Each of the magnetic materials must have a characteristic curve assigned to it. Only material label 3 has a default characteristic associated with it. Practical characteristics may be entered to give realistic solutions. In addition curves may be chosen from the library provided by Vector Fields. The Vector Fields curves are supplied in a directory *$vfdir/bh* where *$vfdir* is the directory into which OPERA was installed (see “Implementation Notes” on page 2-1). See the Reference Manual for information on the BHDATA command. In this example, the default BH curve for material 3 will be used for all of the magnetic regions. The default curve will be stored as *sol.bh* and then loaded back into the program for each magnetic material. To accomplish this:

Select

MODEL ↓
BH or
DE Data → BH/DE editing → Material 3 → Store in file

and enter **sol** as the filename.

`BH or DE data` → `BH/DE editing` → `Material 3` → `Return`

The BH characteristic for material label 3 is shown in Figure 8.8.

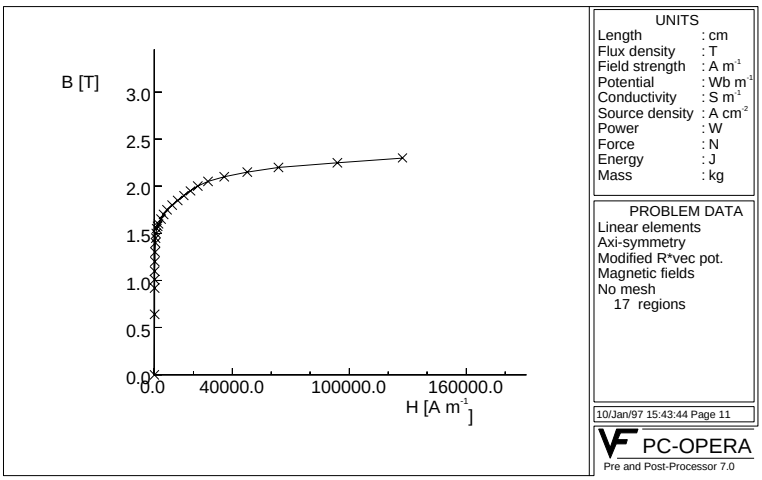


Figure 8.8 The default BH characteristic for material label 3

BH data for material label 5 is now required. The material is selected and the newly created material characteristic file *sol.bh* is loaded so that it is associated with this material.

`BH or DE data` → `BH/DE editing` → `Material 5` → `Load from file`

and select *sol.bh* from the file box.

`BH or DE data` → `BH/DE editing` → `Material 5` → `Return`

Repeat the above procedure for MATERIAL 7 and MATERIAL 9. Then select **`Return`** 3 times to get back to the top level menu.

Setting the Boundary Conditions

Finally, the boundary conditions can be assigned. The display size will be changed as necessary to make setting the boundary conditions easier. The boundary condition **`B normal = 0`** will be set on all external faces, as well as on all faces on the Z-axis (which must have this condition).

Click on

```
DISPLAY ↓
     +mesh (toggles to -mesh)
     Refresh
MODEL ↓
     Boundary Cond. → Vector pot → B normal = 0
```

and select points near each face that lies on the Z axis, i.e.

Inside region 1	0,	0.2
Inside region 5	0,	2
Inside region 6	0,	4
Inside region 7	0,	5.5

Resize the display to set the background region sides

```
Boundary Cond. →Vector pot → Zoom Display → Bounding Box
Boundary Cond. → Vector pot → B normal = 0
```

Inside region 17	0,	20
Inside region 17	0,	-20

and on the far field boundaries at

Inside region 17	25,	40
Inside region 17	25,	-30

and then **Return** to the **MODEL** menu.

Generating the Finite Element Mesh

The finite element mesh is generated as follows:

```
MODEL ↓
     Mesh generator → Generate mesh
```

As mesh generation proceeds the graphics display shows a line drawing of the model. The outer boundaries of the model will be drawn in red with the interior model drawn in white.

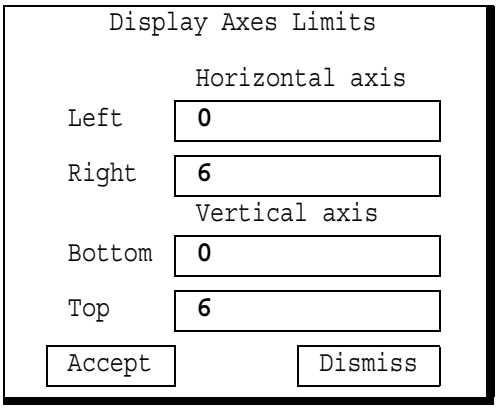
Any red lines in the interior of an electromagnetic model indicate that the mesh is not continuous. These must be corrected before the analysis module is started.

The message box reports on progress and lists the results of checks carried out. The final line should show 0 errors and 0 warnings. Any errors must be corrected, while warnings may need to be corrected. Hit any key to clear the display. Close the menu

MODEL ↓
Mesh generator → **Return**
Return

The mesh can be displayed by

DISPLAY ↓
-mesh ***(toggles to +mesh)***
DISPLAY ↓
Axes limits



and then **Refresh** to show the true aspect ratio of the solenoid, and the mesh created for this model.

Next, the analysis module which is to be used needs to be specified. From the main menu select

FILE ↓
Write file → **Analysis data**

Select the static analysis module (ST) and accept all defaults from the static analysis options by hitting **Return.**

It is now necessary to store the model by selecting

FILE ↓
Write file → **Write model**

entering **sol** as the selected file name and pressing **Accept**.

Analysis

In order to launch the Static Analysis module select

FILE ↓
Start analysis

Within the **Start Analysis** menu select the appropriate filename **(sol)** and **Accept**.

OPERA-2d then reports on the progress of the solution.

On successful completion, the solution process has created a solution file, *sol.st*. This contains the solution data from the static analysis for post processing.

OPERA-2d Post Processor

Reading and displaying the solution

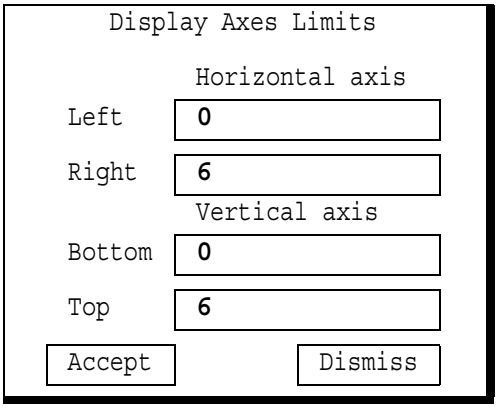
All previous settings within the pre and post processor may be cleared before post processing the model by selecting

OPTIONS ↓
Clear and reset

The file *sol.st* should be read into the post processor as shown in the previous example. A message box is displayed showing the units setting. This can be cleared by hitting any key or mouse button.

To display the model with a 1:1 aspect ratio:

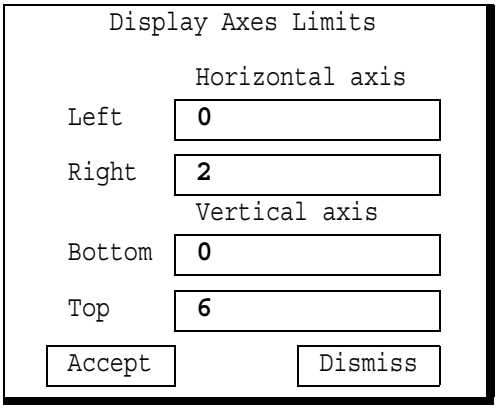
DISPLAY ↓
Axes limits



followed by **Refresh**

In this case a display with an aspect ratio other than 1:1 may be more useful. This will result in the display being distorted but details can be seen more clearly. To do this:

DISPLAY ↓
Axes limits



DISPLAY ↓

Style → **Fill materials**
Style → **Return**
Nodes → **No vertices/nodes**
Style → **Return**
+Labels ***(toggles to -labels)***
Refresh

In addition to changing the size of the display other changes have been made:-

- **Style:** (fill materials) causes the different materials to be outlined instead of each region.
- **Nodes:** (no vertices/nodes) causes no nodes to be displayed.
- **+Labels:** (changes to -Labels) causes no labels to be displayed.

The new display is shown in Figure 8.9

Line Contours of Vector Potential

Before investing a great deal of time in the analysis of any model, the user should verify that the results being analysed are reasonable.

One of the easiest things to check is the flux pattern. The static analysis is solved in terms of a modified vector potential and equipotential contours are equivalent to flux lines in RZ coordinates. To see these contours of modified vector potential:

FIELDS ↓

Contour plot → **Execute**

A message box displaying the model RMS error is given. This can be cleared by hitting any key or mouse button. The equipotential line contours of vector poten-

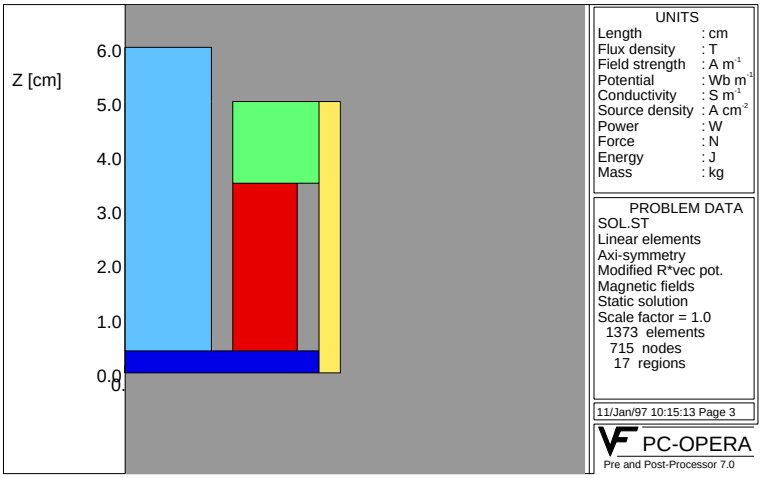


Figure 8.9 The Model Displayed in the post processor

tial over the whole model are displayed as shown in Figure 8.10. These should be checked to ensure that they agree with expected magnetic performance of the solenoid. If errors appear to be present then the model should be modified to correct the error and the analysis re-run.

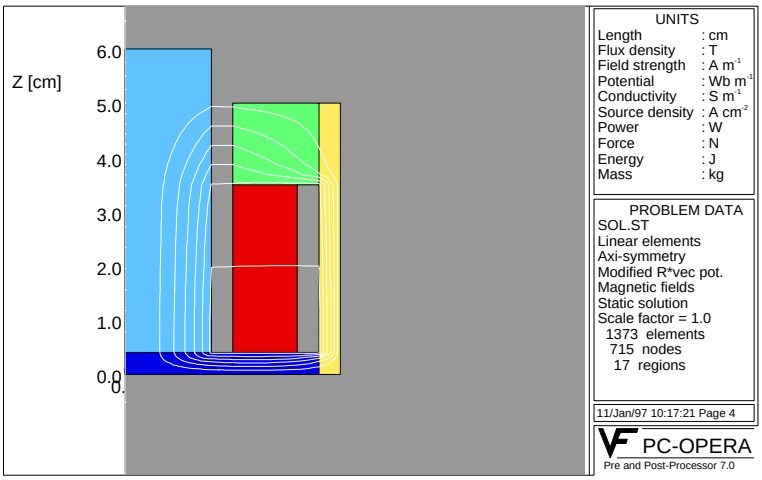


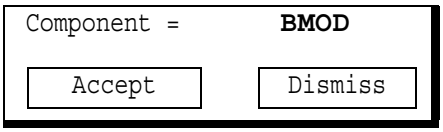
Figure 8.10 Equipotential line contours of modified vector potential - equivalent to lines of magnetic flux

Zone Contours of Flux Density Magnitude

To examine the magnitude of flux density, bmod is selected as the component to be analysed:

FIELDS ↓

Component



FIELDS ↓

Contour plot → Style → Filled zones
Contour plot → Style → Return
Contour plot → No refresh ***(Toggles to Refresh)***
Contour plot → Execute

The picture is redrawn and colour bands of flux density are drawn on the model as well as on the scale at the bottom of the picture as shown in Figure 8.11.

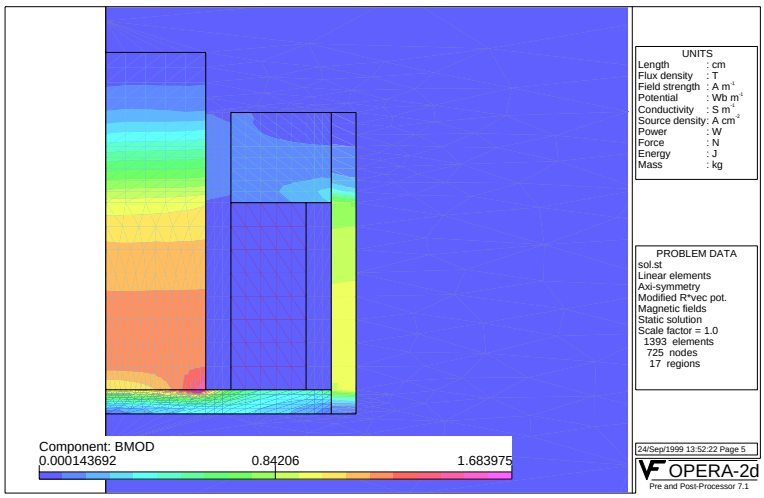


Figure 8.11 Contour plot of the magnitude of flux density

Line Graphs

Graphs representing lines in the two dimensional space may be drawn. The appropriate component needs to be selected prior to drawing the graph. To see a line graph of the Z component of flux density at the base of the plunger (region 5):

FIELDS ↓

Component

Component =	**bz**
Accept	Dismiss

Graphs → **Along line ...**

... options → **line style**

and select **Solid line**

Graphs → **Along line ...**

... options → **Return**

Graphs → **Along line**

Start X coordinate	=	**0**
.....Y coordinate	=	**0.4**
End X coordinate	=	**1.2**
....Y coordinate	=	**0.4**
Curvature of line	=	**0**
Number of points	=	**100**
Accept		Dismiss

The resulting display is shown in Figure 8.12

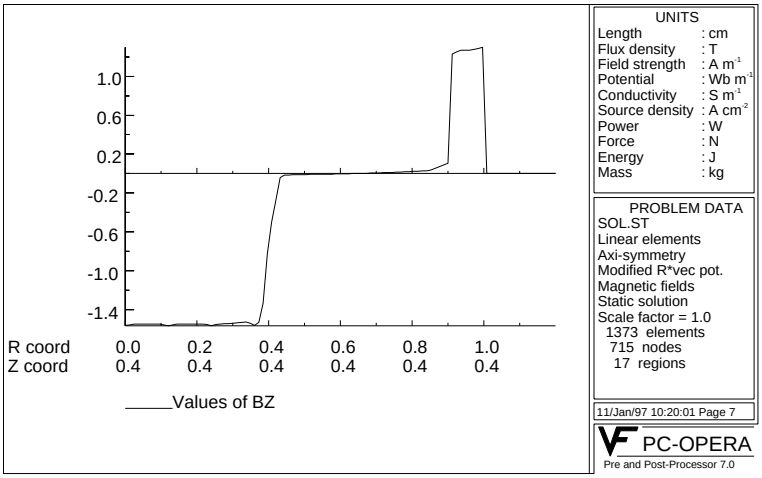


Figure 8.12 A Line Graph of BZ at the base of the Plunger

Line Integration

To calculate the total force on a body, in this case the force on the plunger (regions 5 to 7), an integration surface around the body is defined. Working in two dimensional space, OPERA-2d calculates the total force on a body by performing a series of line integrals encircling the body. The line integrals must be in an anticlockwise direction.

In this example, the first line would be:

FIELDS ↓

Integrals → **Along line**

Start X coordinate	=	**0**
......Y coordinate	=	**0.4**
End X coordinate	=	**0.45**
....Y coordinate	=	**0.4**
Curvature of line	=	**0**
Error(<1) or No. of steps(>1)	=	**0.01**

Accept Dismiss

A message box displays

```
WARNING Integration path is not all in air
```

and indicates that the line integral is taken right on the interface between regions 1 and 5. The program correctly performs the integration by evaluating the Maxwell stresses as if the line is a vanishingly narrow cut in the material. **It is recommended that line integrals be performed in air regions where possible, for improved accuracy**. Hit any key to clear the message box. The integration results appears in a second message box and this can also be cleared.

To find the total force, two more line integrals must be performed and the results (integral of z-component of Maxwell Stress) must be added together. The addition can be automatically performed by making use of the ACCUMULATOR feature in the Line integral option which adds the results to a total after each line integral is completed. The last two line integrals are carried out in this way:

FIELDS ↓
Integrals → .....Options
(Under line integral) **→ Accumulate → Add**
Integrals → .....Options
(Under line integral) **→ Accumulate → Return**
Integrals → .....Options
(Under line integral) **→ Return**

FIELDS ↓
Integrals → Along line

Start X coordinate	=	**0.45**
......Y coordinate	=	**0.4**
End X coordinate	=	**0.45**
....Y coordinate	=	**7**
Curvature of line	=	**0**
Error(<1) or No. of steps(>1)	=	**0.01**

Accept Dismiss

FIELDS ↓
Integrals → Along line

Start X coordinate	=	**0.45**
......Y coordinate	=	**6.05**
End X coordinate	=	**0**
....Y coordinate	=	**7**
Curvature of line	=	**0**
Error(<1) or No. of steps(>1)	=	**0.01**

Accept Dismiss

The integral of Maxwell stress is shown each time the line integral is used as well as the accumulated integrals when the accumulator is used. It is not necessary to integrate along the Z-axis since the three integrals describe a cylinder that completely surrounds the plunger.

To end the current session now and close the pre and post processor:

FILE ↓
End OPERA-2d/pp

Chapter 9
A Permanent Magnet Motor Example

Introduction

This example of a simplified six pole permanent magnet synchronous motor will illustrate the use of several features, some of which have not been included in previous examples. They are:

- Using model symmetry
- Copy command
- Replication feature
- Permanent magnets
- Symmetry boundary conditions
- BH curves
- Quadratic elements

Since the motor has rotational periodicity (symmetry), only a single sector needs to be modelled (in this case - 60 degrees). This reduces the amount of work needed to create the model and reduces the number of elements and nodes required. Figure 9.1 shows the complete model.

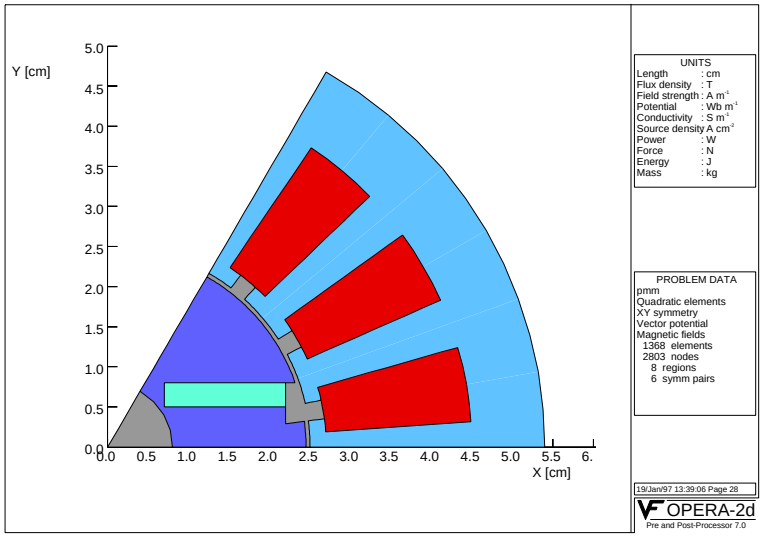


Figure 9.1 The complete model

Pre Processing

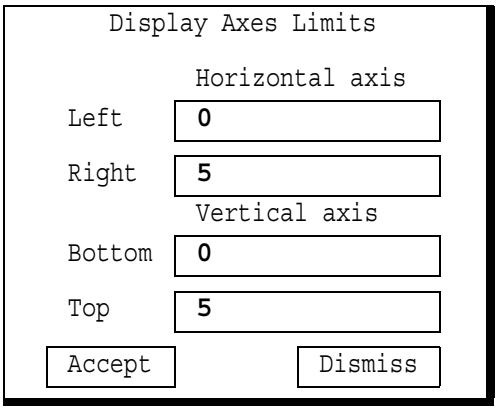
Setting the Pre and Post Processor Environment

Launch OPERA-2d and enter the pre and post processor as described in previous tutorials. Change the default S.I. units to a more suitable set.

UNITS ↓
Length unit → Centimetre
Length unit → Return
Density unit → Amps/cm2**
Density unit → Return
Return

The axes dimensions also need to be set to be suitable for a stator radius of 5 cm. To do this:

DISPLAY ↓
Axes limits



Refresh

Model information is given on the right hand side of the graphics display. It can be seen that the default element type is linear. It is recommended to solve a problem with linear elements initially, so that the model solution can be validated. After this, quadratic elements can be used to obtain an improved solution accepting the associated increase in computational effort and time.

However in this example, quadratic elements will be used to demonstrate their application. This will reduce the RMS error and produce more accurate results. Select

MODEL ↓

Solution Type → **Quad elements (Linear elements)** ***toggle***

and the menu item will toggle to **Quad elements**. Then

Solution Type → **Return**

The material label to be used when drawing the stator regions will now be set. Material label 5 and a permeability of 2500 will be used, although material characteristics may easily be modified at a later stage.

Draw regions → **Region defaults ...**
... material type

Material label	=	**5**
Mu or epsilon	=	**2500**
Density	=	**0**
Conductivity	=	**0**
Phase/angle	=	**0**
Velocity	=	**0**

Accept Dismiss

Entering Construction Lines

To outline the model geometry, construction lines may be used. These are drawn by selecting

Draw regions → **New region ...**
... Polygon → **Enter C_lines** → **Line**

and then completing the box as shown below.

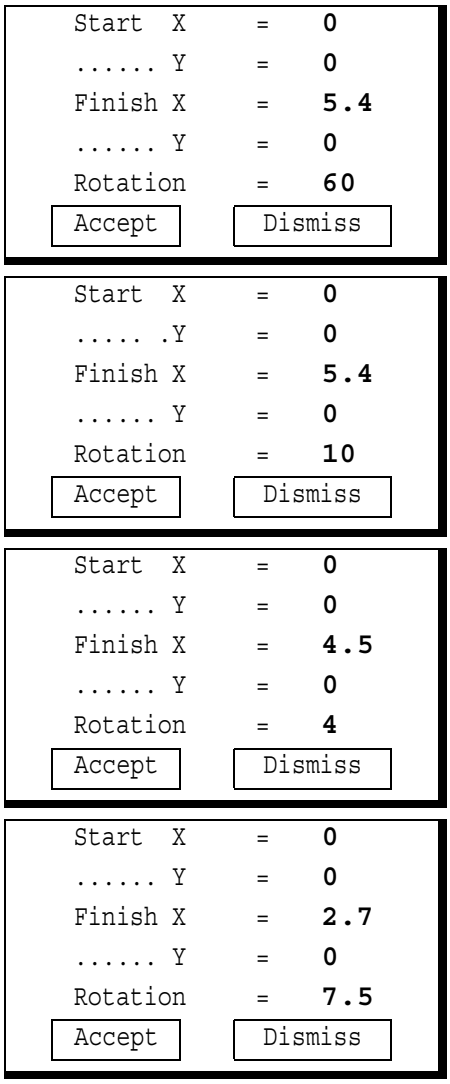
Start X	=	**0**
..... Y	=	**0**
Finish X	=	**5.4**
..... Y	=	**0**
Rotation	=	**0**

Accept Dismiss

and the construction line will appear on the screen from (0,0) to (5.4,0).

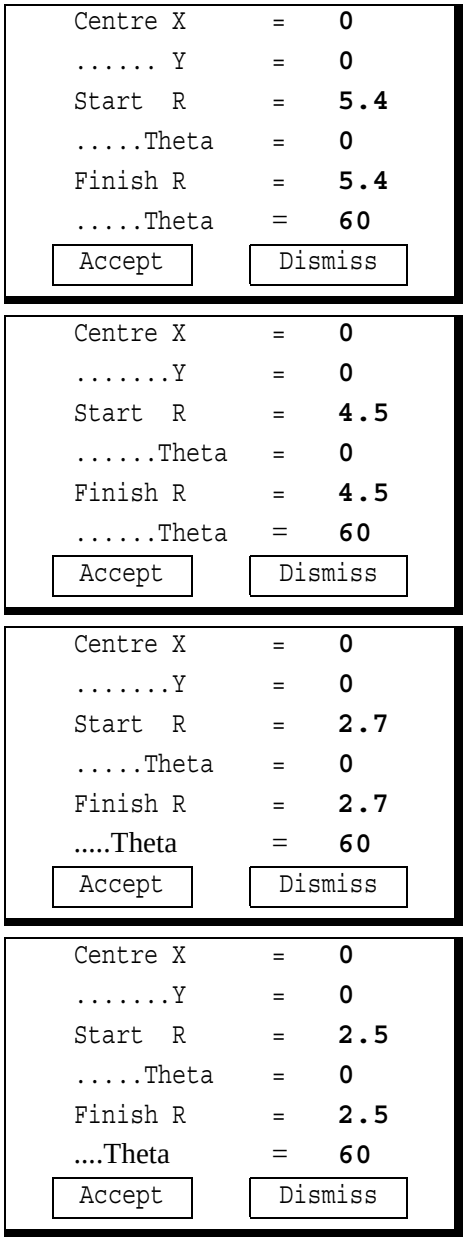
Construction lines do not form part of the model but are useful for defining points within the geometry.

Several construction lines are needed so the following lines should be added in the same fashion.



Arced construction lines can be added by selecting

Draw regions → New region ...
... Polygon → Enter C_lines → Arc



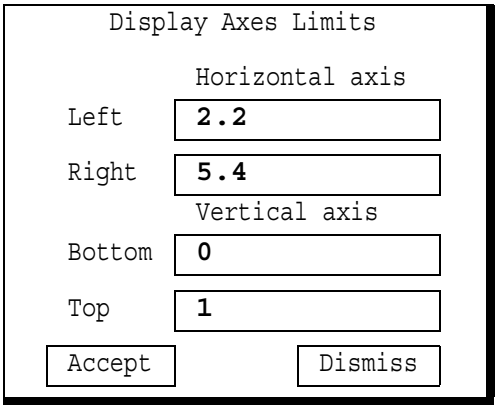
Draw regions →New Region ...
... polygon → Enter C_lines → Return

Drawing the Stator Tooth

To help draw the stator, the display will be resized by

MODEL ↓

Zoom display → Numerical axes limits



The stator tooth will be constructed by connecting the region vertices. These can be identified with intersection points of the construction lines. Figure 9.2 shows region 1 completed.

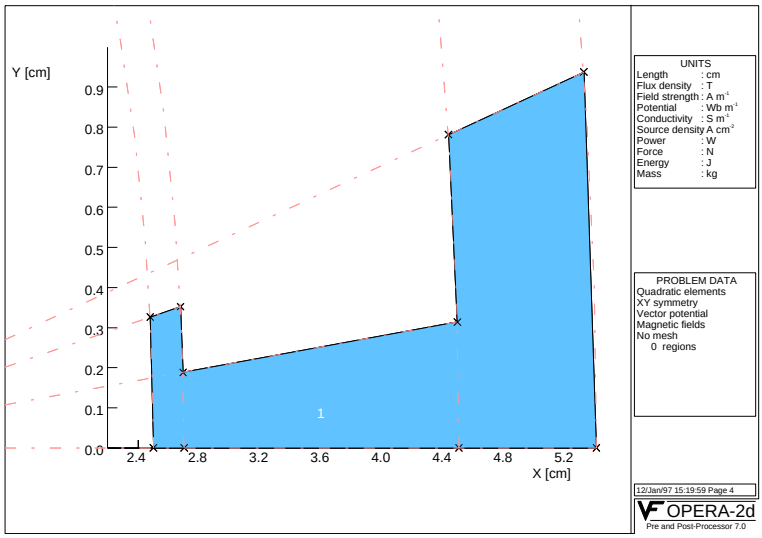


Figure 9.2 The first region - Half the stator tooth

Draw regions → New region ...

... Polygon → Mouse Input → At Intersection

Use the mouse to select the intersections of construction lines near the following points (you may need to temporarily hide the menus by pressing **F1**).

5.4, 0
4.5, 0
2.7, 0
2.5, 0
2.5, 0.3
2.7, 0.3
2.7, 0.2
4.5, 0.3
4.5, 0.6
5.4, 0.8

```
Draw regions → New region ...
                  ... Polygon → Mouse Input → Close polygon
Draw regions → New region ...
                  ... Polygon → Mouse Input → Return
Draw regions →  New region ...
                  ... Polygon → Return
Draw regions → Return
```

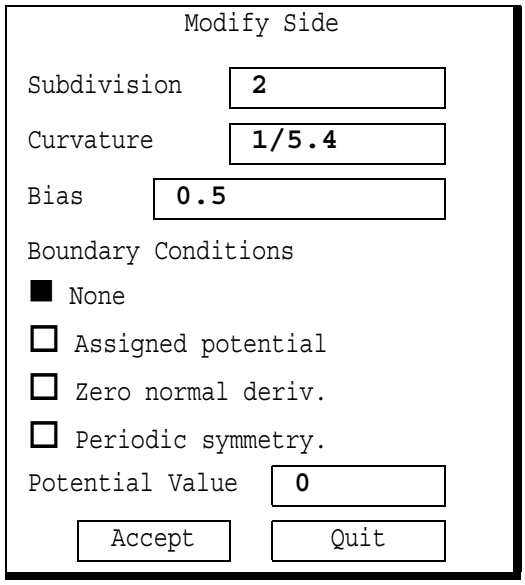
It is more efficient to set the correct number of subdivisions and curvature of sides, before any copies are made. To modify region 1, click on

```
MODEL ↓
     Modify regions → Modify sides ...
                        ... by picking → All properties...
                                           ... of one side
```

and select the side by clicking the mouse inside region 1 at

5.3, 0.4

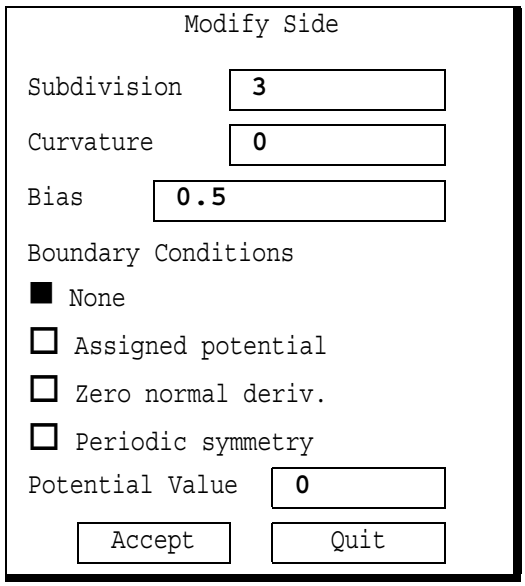
and fill in the box as shown below.



Continue modifying the other sides of region 1 as shown below.

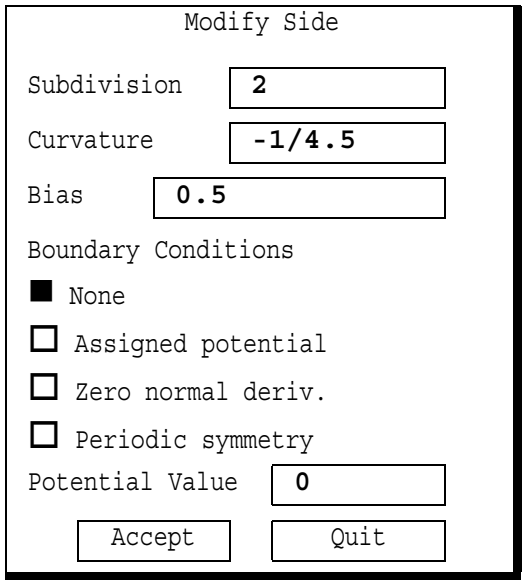
MODEL ↓
Modify regions → **Modify sides ...**
... by picking → **All properties...**
... of one side

mouse click at: 5, 0.7



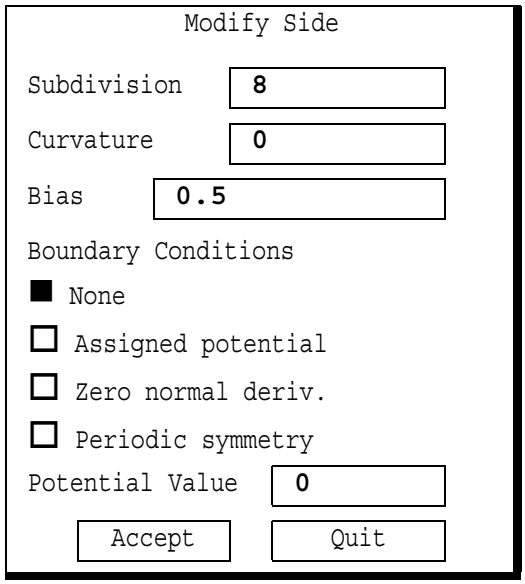
MODEL ↓
Modify regions → Modify sides ...
... by picking → All properties...
... of one side

mouse click at: 4.5, 0.4



MODEL ↓
Modify regions → Modify sides ...
... by picking → All properties...
... of one side

mouse click at: 3.5, 0.2



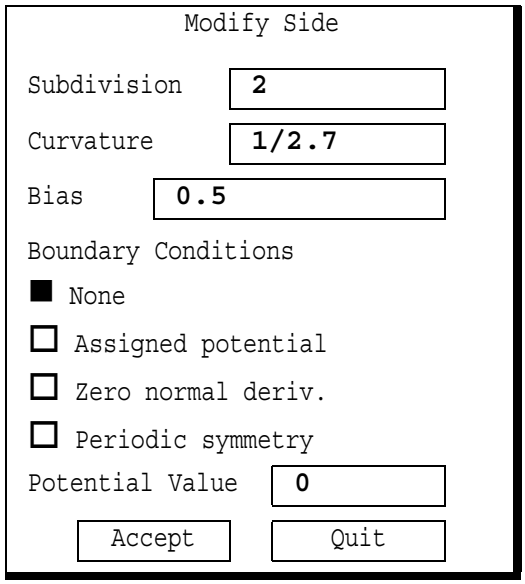
MODEL ↓

Modify regions → **Modify sides ...**

... by picking → **All properties...**

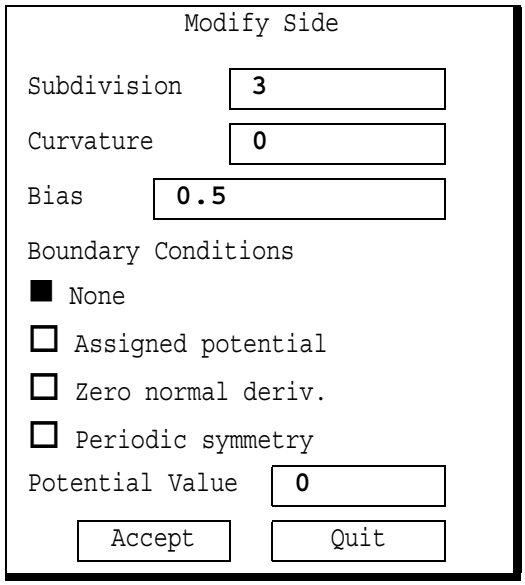
... of one side

mouse click at: 2.65, 0.25



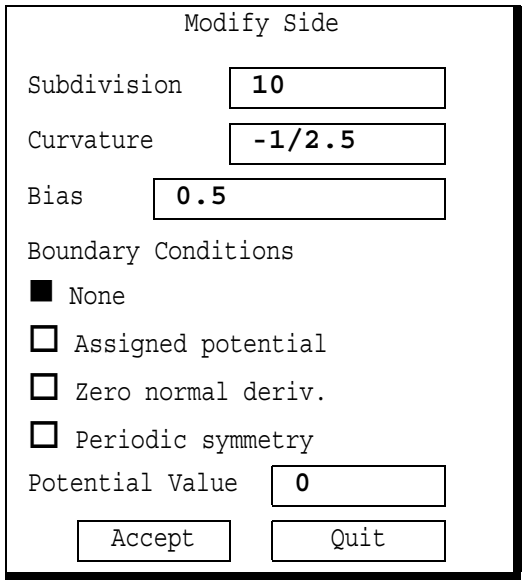
MODEL ↓
Modify regions → Modify sides...
... by picking → All properties...
... of one side

mouse click at: 2.55, 0.3



MODEL ↓
Modify regions → Modify sides → All properties...
... of one side

mouse click at: 2.5, 0.2



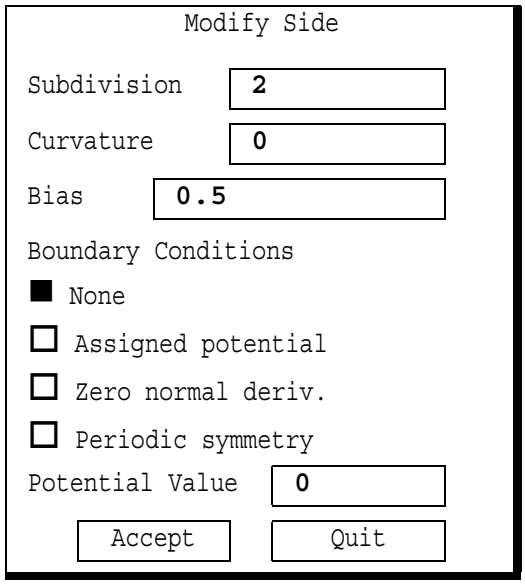
MODEL ↓

Modify regions → **Modify sides...**

... by picking → **All properties...**

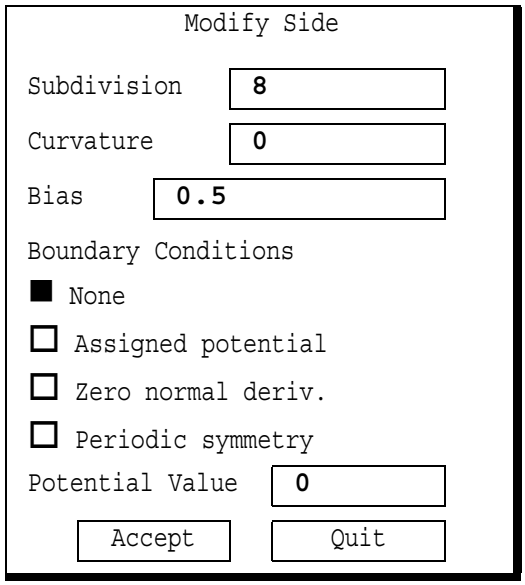
... of one side

mouse click at: 2.6, 0



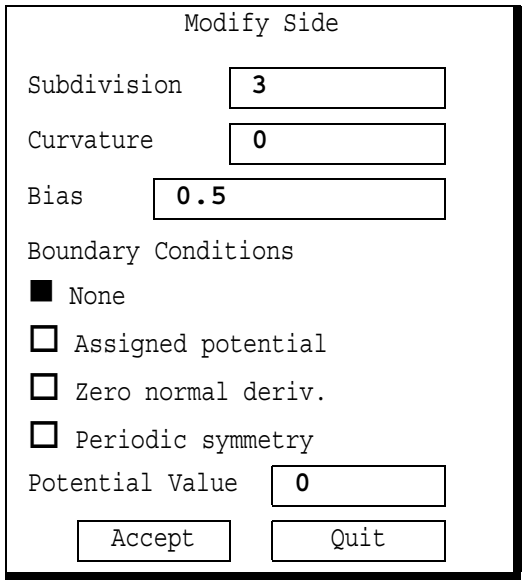
MODEL ↓
Modify regions → Modify sides ...
... by picking → All properties...
...of one side

mouse click at: 3.5, 0



MODEL ↓
Modify regions → Modify sides ...
... by picking → All properties...
... of one side

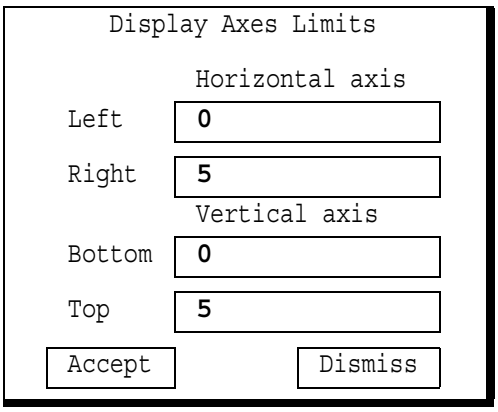
mouse click at: 5, 0



Modify regions → Modify sides ...
... by picking → Return
Modify regions → Return
Return

To see the mesh that has been created in region 1 click on

DISPLAY ↓
-Mesh *(toggles to +mesh)*
Axes limits



Refresh

The display including the mesh is shown in Figure 9.3.

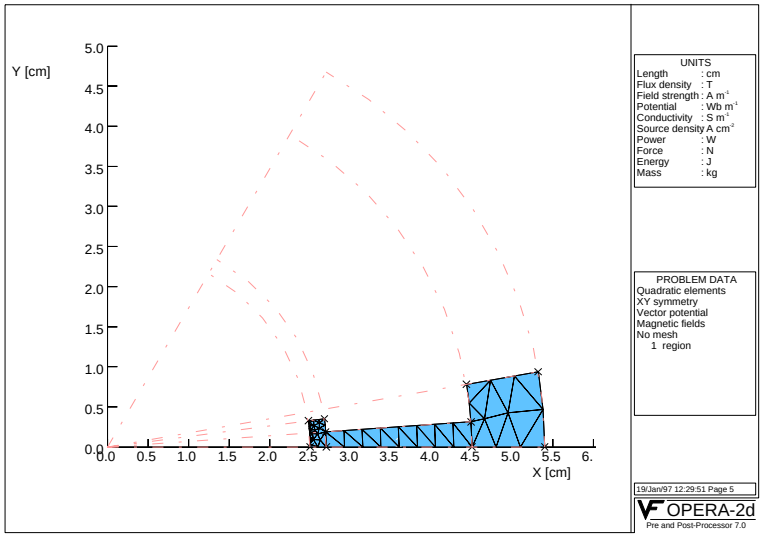


Figure 9.3 The display of the mesh for region 1

Using Copies and Replications

OPERA-2d allows two distinct methods of producing new regions with the same geometry as existing regions.

These are:-

- Copy

  This produces a new region (with a new number). Once created, the new region may be modified or deleted without affecting the region from which it was copied.

- Replication

  This produces a clone of the original region. Any modifications (including deletion) will affect the region from which it was replicated.

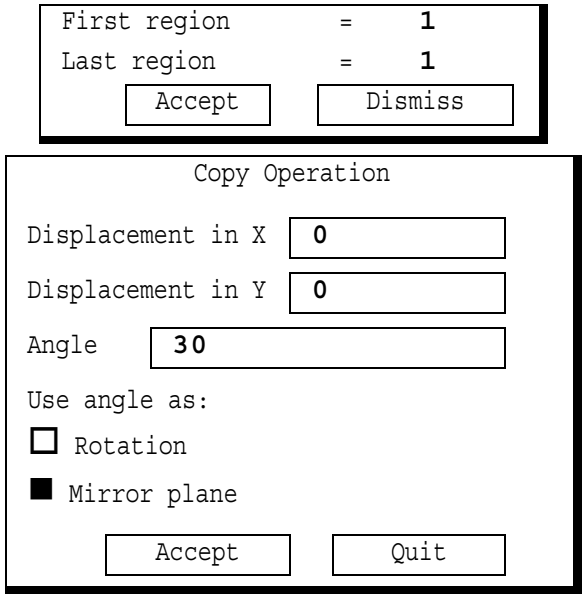
The remainder of the stator core will be made by making copies and replications of region 1.

It is possible to produce all the regions of the stator core geometry from region 1, by replication. However it is not permitted to set periodic boundary conditions on a region with replications. Hence the two regions (region 1 and a copy, region 2) which will take the periodic boundary condition, will be created as separate regions. Region 3, another copy of region 1, is used for replications.

To create region 2, a copy of region 1 is made. Select

MODEL ↓

Copy regions → Region numbers



Regions 1 and 2 are shown in Figure 9.4.

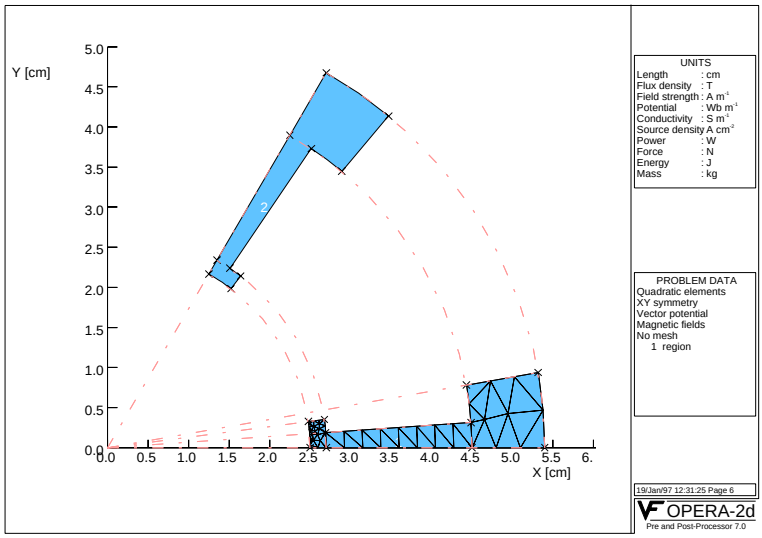
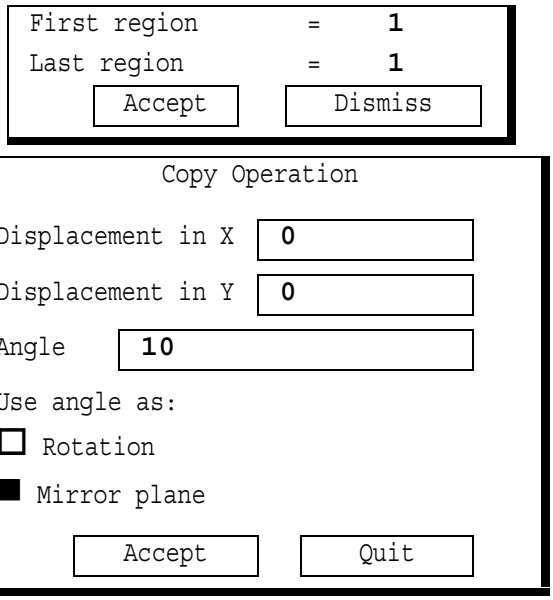


Figure 9.4 Regions 1 and 2 of the stator core

To create region 3, another copy of region 1 is made:

Copy regions → Region numbers



The remaining parts of the stator core will be created as replications of region 3. Select

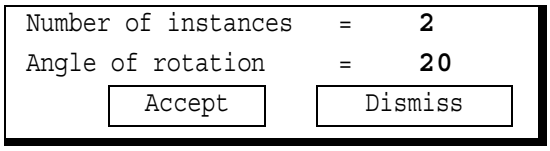
Modify regions → Modify region → Pick region

an select region 3 by clicking the mouse at coordinate:

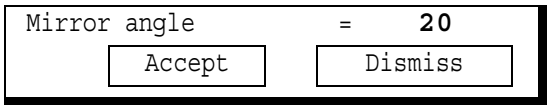
5, 1

Then set the replication data for region 3 by choosing

Modify region → Pick region → Replication → Rotational sym.



Modify region → Pick region → Replication → Mirror on



Modify region → Pick region → Replication → Return
Modify regions →Modify region → Pick region → Return
Modify regions → Modify region → Return
Modify regions → Return

The complete stator is drawn as shown in Figure 9.5. Any modifications made to the replicated regions affect the original region 3. Similarly, any changes made to the original region 3 will affect the replications.

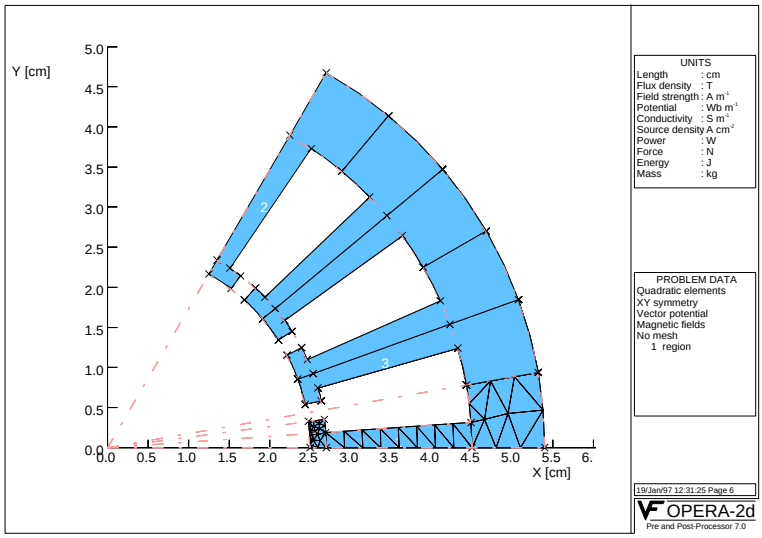


Figure 9.5 The stator core - Copies and replications of region 1

Drawing the Conductor Regions

The three conductors will be drawn from one region with two replications. This means that each slot will carry the same current. The material properties will be given a material label of 1 with a permeability of 1. These are set by:

Draw regions → Region defaults ...

... material type

Material label	=	1
Mu or epsilon	=	1
Density	=	0
Conductivity	=	0
Phase/angle	=	0
Velocity	=	0

Accept Dismiss

Then the region may be drawn:

Draw regions →New region ...
... Polygon → Mouse input → At old point
4.5, 1
4.5, 0.6
4.5, 0.2
2.7, 0.2
2.7, 0.3
2.6, 0.6
2.6, 0.8
Draw regions → New region ...
... Polygon → Mouse input → Close polygon

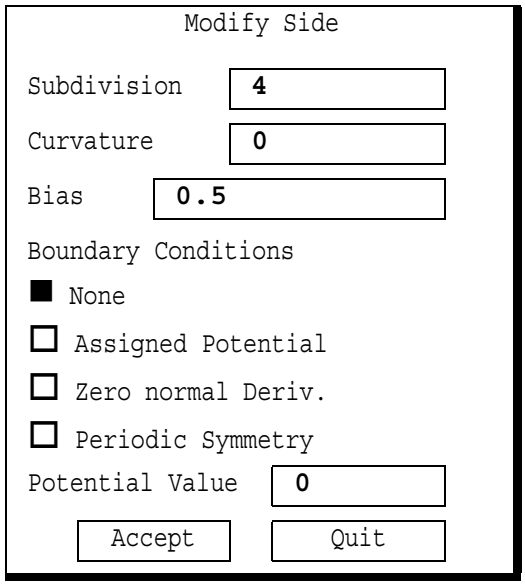
Region 4 is created.

Draw regions → New region ...
... Polygon → Mouse input → Return
Draw regions → New region ...
... Polygon → Return
Draw regions → Return

The number of subdivisions along one of the sides will be changed in region 4, to improve the mesh click on

MODEL ↓
Modify regions → Modify sides ...
... by picking → All properties...
... of one side

Click mouse inside region 4 near (2.7,0.5)



Modify regions → Modify sides...
... by picking → Return

Since the area of region 4 is known at this stage, it can be used to define a current density equivalent to a total current of 400 A. Modifications to region 4 will involve changing this current density and setting the replications.

Modify regions → Modify Region → Pick Region

Click mouse inside region 4 near (2.7,0.5)

Modify regions →Modify Region → Pick region → Material data

Material label	=	1
Mu or epsilon	=	1
Density	=	400/area
Conductivity	=	0
Phase/angle	=	0
Velocity	=	0

Accept Dismiss

Now modify the replication data of this region by

Modify region → Pick Region → replication → Rotational sym.

Number of instances	=	3
Angle of rotation	=	20

Accept Dismiss

Modify region → Pick Region → Replication → Return
Modify regions →Modify Region → Pick Region → Return
Modify regions → Modify Region → Return
Modify regions → Return

The result of this is shown in Figure 9.6

Drawing the Rotor

Construction lines will be used to outline the geometry of the rotor. Resizing the display ready to input the rotor:

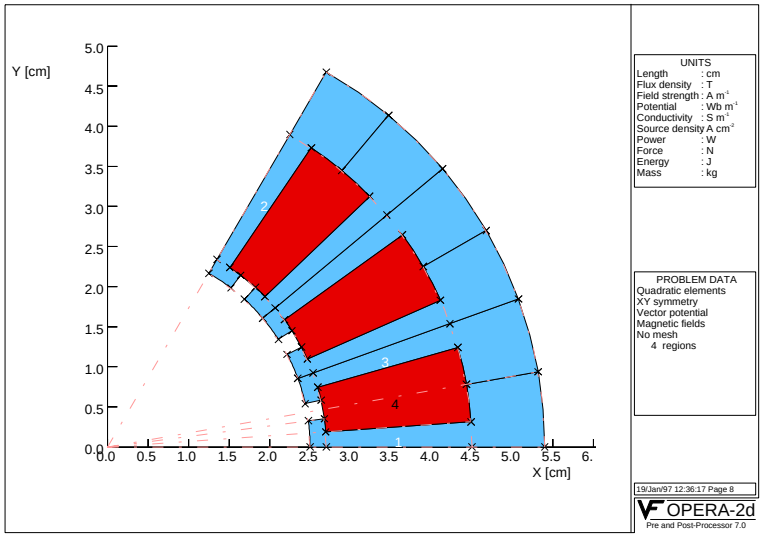
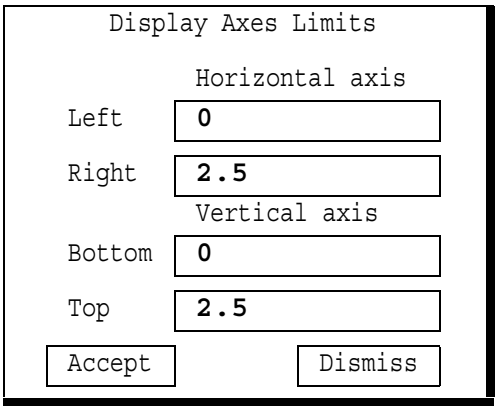


Figure 9.6 The complete stator - Including conductors

MODEL ↓

Zoom display → Numerical axes limits



Enter the construction lines

Draw regions →New region ...

... Polygon → Enter C_lines → Arc

Centre X	=	0
.....Y	=	0
Start R	=	2.45
.....Theta	=	0
Finish R	=	2.45
.....Theta	=	60

Accept Dismiss

Centre X	=	0
.....Y	=	0
Start R	=	0.8
.....Theta	=	0
Finish R	=	0.8
.....Theta	=	60

Accept Dismiss

Draw regions → New region...

... Polygon → Enter C_lines → Line

Start X	=	2.2
.....Y	=	0
End X	=	2.2
....Y	=	0.8
Rotation	=	0

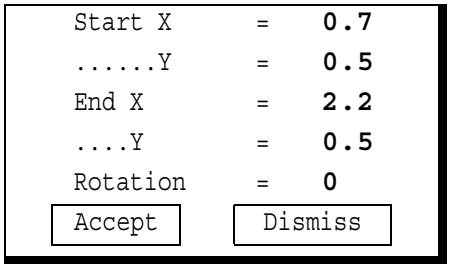
Accept Dismiss

Start X	=	0.7
.....Y	=	0
End X	=	0.7
....Y	=	0.8
Rotation	=	0

Accept Dismiss

Start X	=	0.7
.....Y	=	0.8
End X	=	2.45
....Y	=	0.8
Rotation	=	0

Accept Dismiss



Draw regions →New region...
... Polygon → Enter C_lines → Return

Since this is a permanent magnet motor, the rotor has two material labels associated with it, the rotor steel and the permanent magnet. The permanent magnet will be created by:

Draw regions →New region ...
... Polygon → Mouse input → At intersection

and select the construction line intersections at

2.2, 0.5
2.2, 0.8
0.7, 0.8
0.7, 0.5

Draw regions →New region ...
... Polygon → Mouse input → Close polygon

Region 5 is created.

The pre and post processor retains the material characteristics chosen previously and this is confirmed by the permanent magnet region being drawn in red. The region material will be changed to a hard magnetic material (permanent magnet) later.

Drawing the rotor core select

Draw regions →New region...
... Polygon → Mouse input → At intersection

2.45, 0
2.4, 0.3
2.2, 0.25
2.2, 0.5
0.7, 0.5
0.7, 0.8
2.2, 0.8
2.3, 0.8
1.2, 2.1

0.4, 0.7
0.8, 0
Draw regions → New region ...
... Polygon → Mouse input → Close polygon

Region 6 is created.

The next section to be created is the stainless steel shaft. Since a magnetostatic solution is required and the shaft is non-magnetic, it may be represented as an air region. Begin by:

Draw regions → New region ...
... Polygon → Mouse input → At intersection

0.8, 0
0.4, 0.7
0, 0

Draw regions → New region ...
... Polygon → Mouse input → Close polygon

Region 7 is created.

Draw regions → New region ...
... Polygon → Mouse input → Return
Draw regions → New region ...
... Polygon → Return
Draw regions → Return

The rotor section has straight region sides instead of the expected curved lines, in addition to having the incorrect material characteristics, as can be seen in Figure 9.7

These will be changed by modifying the regions. The material characteristics for regions 5 to 7 may be changed as follows:

Modify regions → Modify Region → Pick Region

Click mouse inside region 5 near (1.6,0.6)

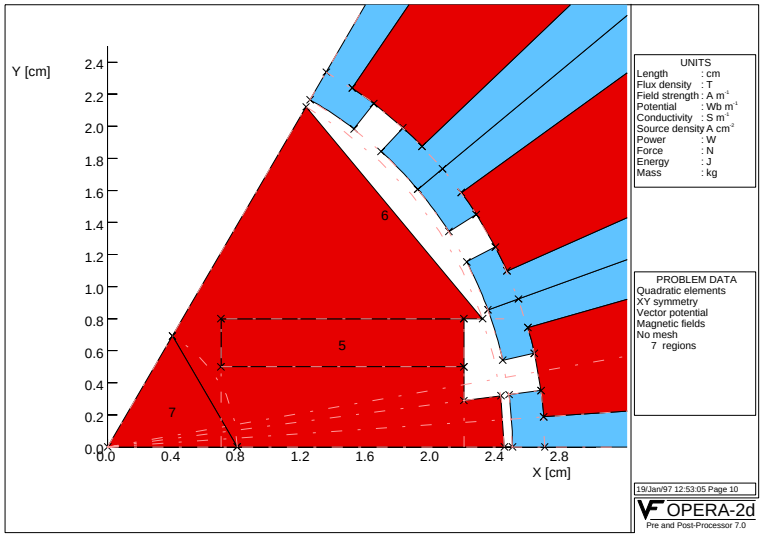


Figure 9.7 The rotor requires further modification

Modify regions → Modify Region → Pick Region → Material data

Material label	=	6
Mu or epsilon	=	1.1
Density	=	0
Conductivity	=	0
Phase/angle	=	90
Velocity	=	0

Accept Dismiss

Modify regions → Modify region → Pick Region → Return

The Phase/angle parameter in the Material data specifies the magnetisation direction of the material.

Modify regions → Modify Region → Pick Region

Click mouse inside region 6 near (1.5,1.2)

Modify regions →Modify Region → Pick Region → Material data

Material label	=	**4**
Mu or epsilon	=	**2500**
Density	=	**0**
Conductivity	=	**0**
Phase/angle	=	**0**
Velocity	=	**0**

Accept Dismiss

Modify regions →Modify Region → Pick Region → Return

Modify regions → Modify Region → Pick Region

Click mouse inside region 7 near (0.4,0.3)

Modify regions →Modify Region → Pick Region → Material data

Material label	=	**0**
Mu or epsilon	=	**1**
Density	=	**0**
Conductivity	=	**0**
Phase/angle	=	**0**
Velocity	=	**0**

Accept Dismiss

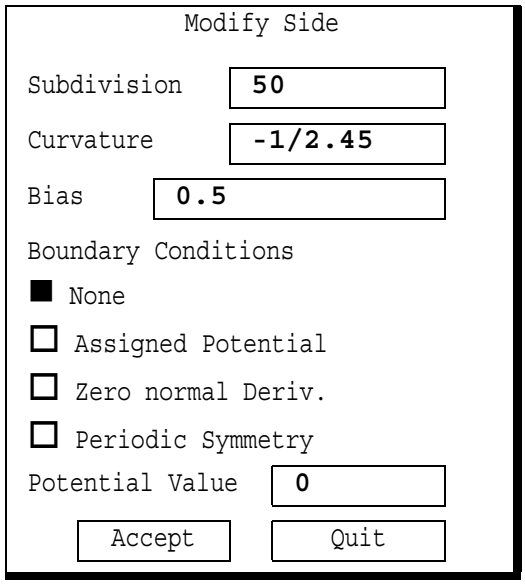
Modify regions →Modify Region → Pick Region → Return
Modify regions → Modify Region → Return

The region subdivisions and line curvatures will now be set in regions 5, 6, and 7. The line curvature changes may be seen as they are made but the subdivision change will only be noticed when the mesh is displayed.

To make the changes:

MODEL ↓
Modify regions → Modify sides ...
... by picking → All properties...
... of one side

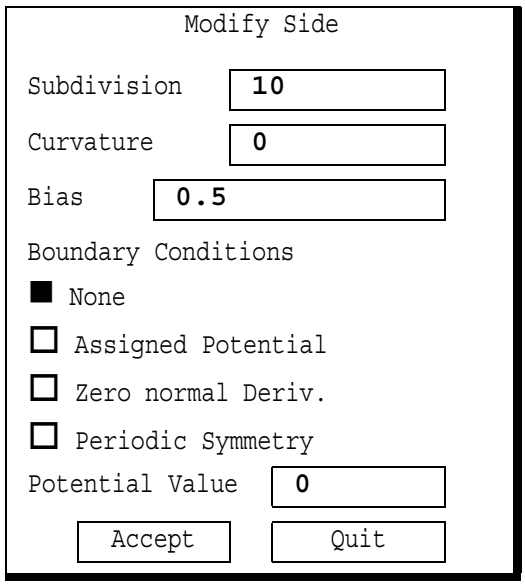
Click mouse inside region 6 near (1.6,1.4)



Continue making the modifications listed below.

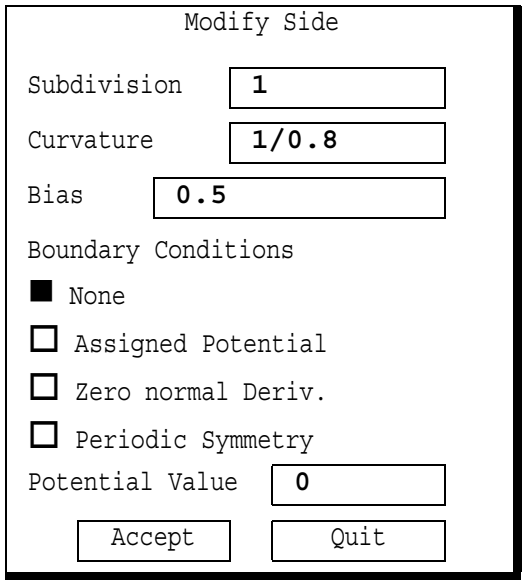
MODEL ↓
Modify regions → **Modify sides ...**
... by picking → **All properties...**
... of one side

Click mouse inside region 6 near (1.0,1.4)



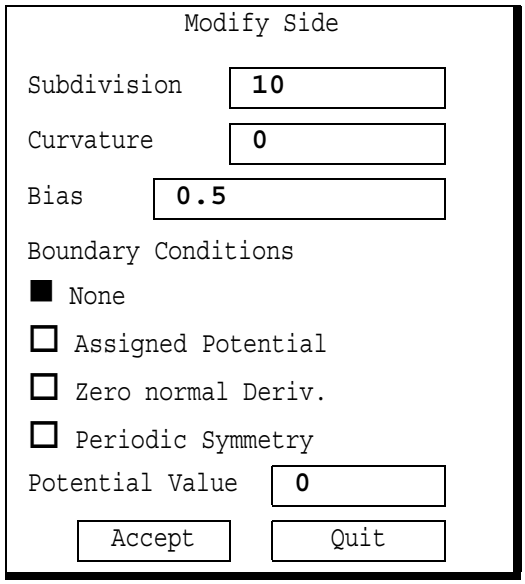
MODEL ↓
Modify regions → Modify sides ...
... by picking → All properties...
... of one side

Click mouse inside region 6 near (0.6,0.4)



MODEL ↓
Modify regions → Modify sides ...
... by picking → All properties...
... of one side

Click mouse inside region 6 near (1.4,0.1)



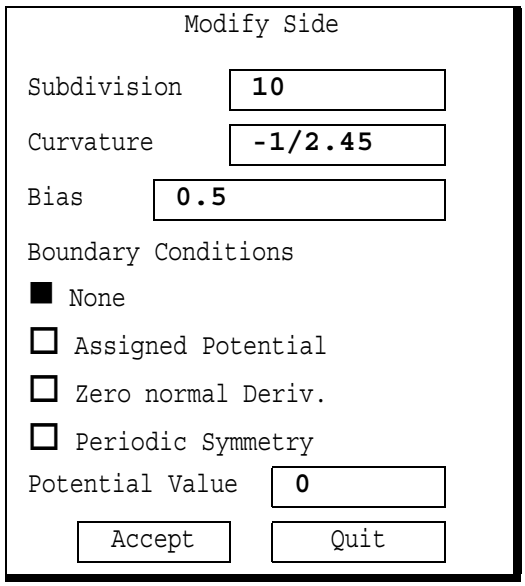
MODEL ↓

Modify regions → **Modify sides ...**

... by picking → **All properties...**

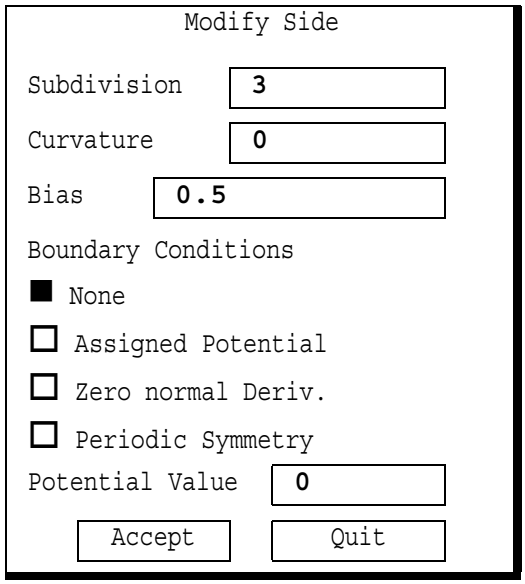
... of one side

Click mouse inside region 6 near (2.4,0.2)



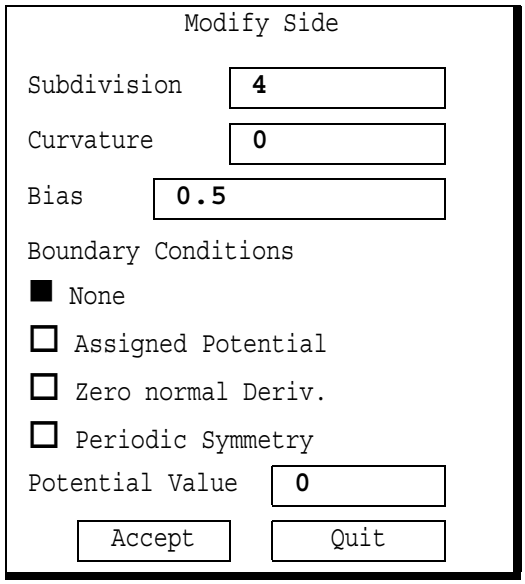
MODEL ↓
Modify regions → **Modify sides...**
... by picking → **All properties...**
... of one side

Click mouse inside region 6 near (2.3,0.3)



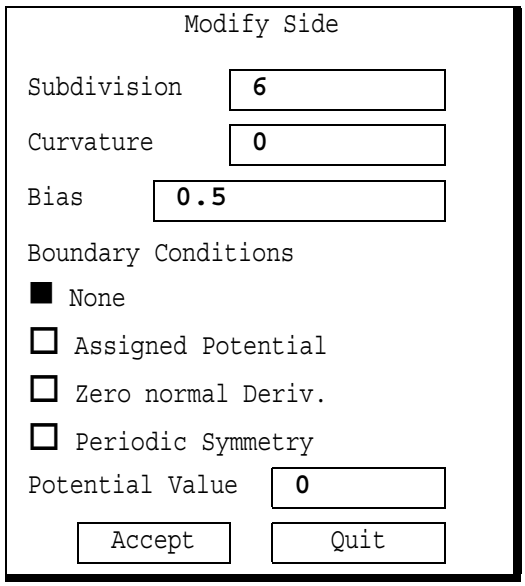
MODEL ↓
Modify regions → **Modify sides...**
... by picking → **All properties...**
... of one side

Click mouse inside region 6 near (2.2,0.4)



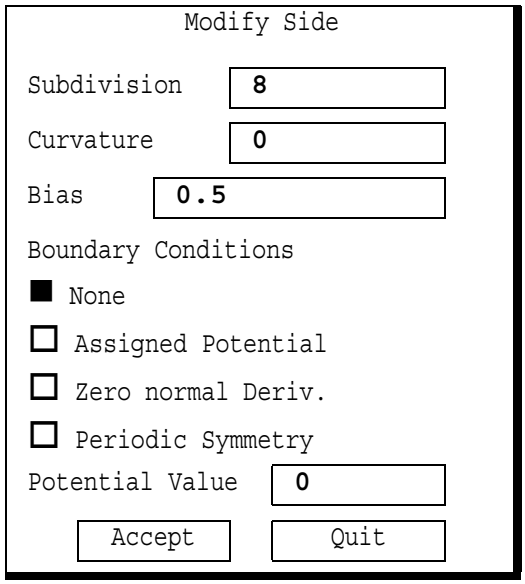
MODEL ↓
Modify regions → **Modify sides...**
... by picking → **All properties...**
... of one side

Click mouse inside region 5 near (2.2,0.6)



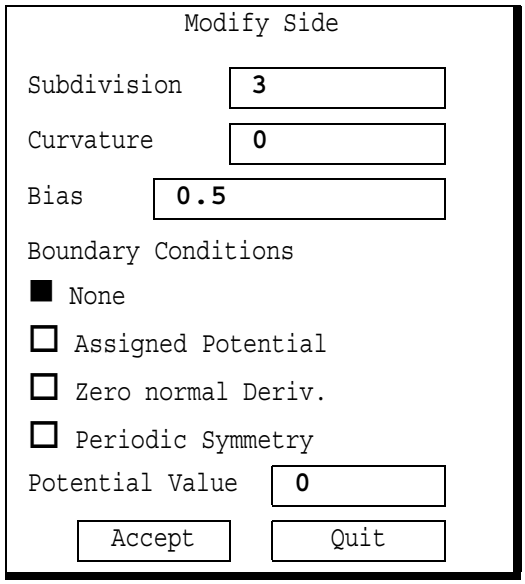
MODEL ↓
Modify regions → **Modify sides...**
... by picking → **All properties...**
... of one side

Click mouse inside region 5 near (1.6,0.5)



MODEL ↓
Modify regions → **Modify sides...**
... by picking → **All properties...**
... of one side

Click mouse inside region 5 near (0.8,0.6)



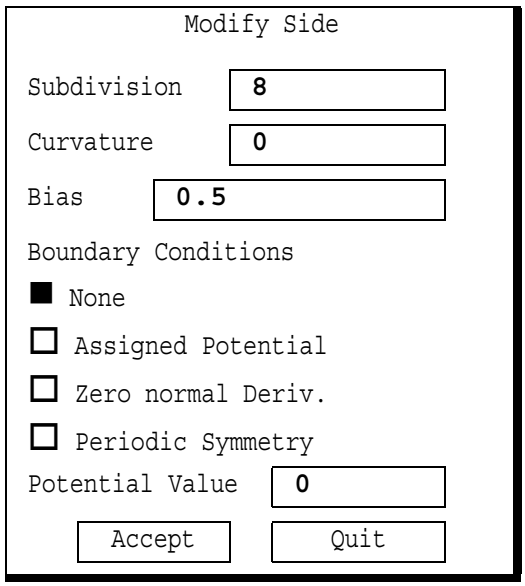
MODEL ↓

Modify regions → **Modify sides...**

... by picking → **All properties...**

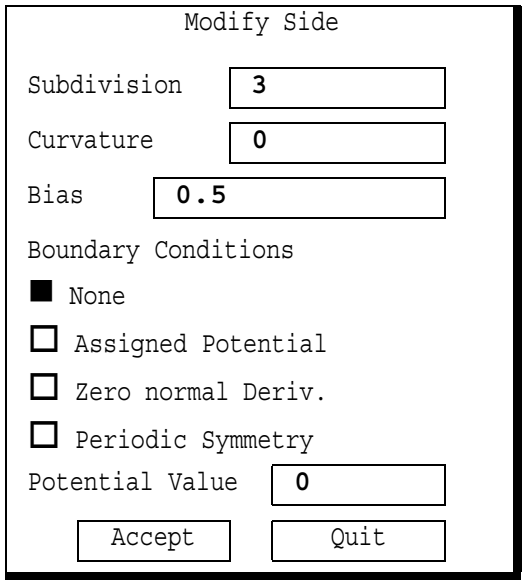
...of one side

Click mouse inside region 5 near (1.6,0.8)



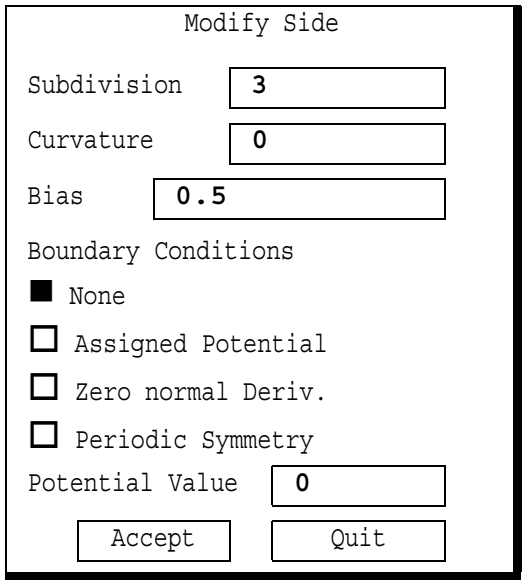
MODEL ↓
Modify regions → **Modify sides...**
... by picking → **All properties...**
... of one side

Click mouse inside region 7 near (0.3,0.4)



MODEL ↓
Modify regions → **Modify sides...**
... by picking → **All properties...**
... of one side

Click mouse inside region 7 near (0.4,0)



Modify regions → Modify sides...
...by picking → Return
Modify regions → Return
Return

To re-display with the mesh shown:

DISPLAY ↓
-Mesh (toggles to +Mesh)

DISPLAY ↓
refresh

Regions 5, 6, and 7 are displayed with the finite element mesh as shown in Figure 9.8. The elements in region 5 are quite large in comparison to the surrounding regions. This is a characteristic of polygon regions such as region 5 which have a large (poor) aspect ratio.

There are several ways to improve the mesh in region 5. Examples include:-

- Increasing the number of elements on the edges of the region. This is not very effective for regions with a poor aspect ratio.
- Subdividing the region into two or more regions. This requires modifying the region geometry or deleting the region to allow two or more regions to be drawn.
- Changing the shape code of the region.

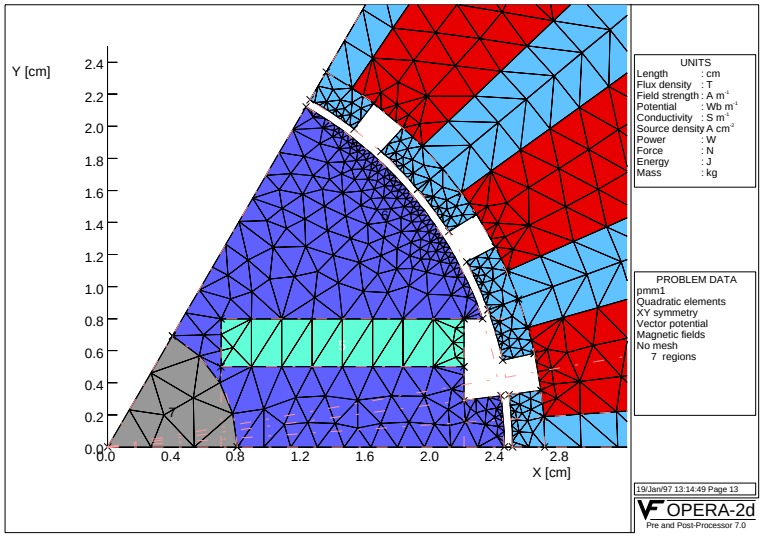
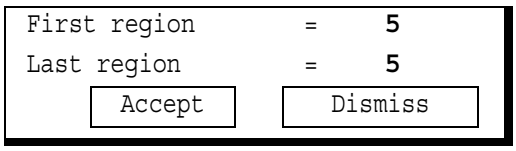


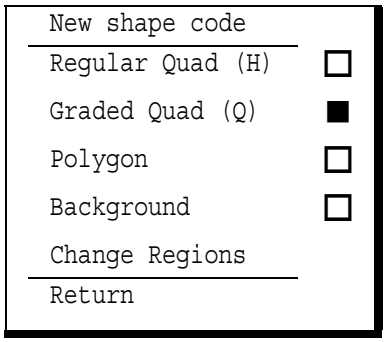
Figure 9.8 The rotor including finite element mesh

In this case, region 5 will be changed from a polygon shaped region (the default) to a shape Q region. Do this by:

MODEL ↓

Change regions → **Region numbers**





and select **Graded quad (Q)** followed by **Change regions**

By redrawing the display the effect of changing the shape of region 5 may be seen:

Zoom display → **Same size**

The final region to be created is the arguable between the rotor and the stator. First set the properties of the region to be drawn by

MODEL ↓

Draw regions → Region defaults ...

... material type

Material label	=	**0**
Mu or epsilon	=	**1**
Density	=	**0**
Conductivity	=	**0**
Phase/angle	=	**0**
Velocity	=	**0**

Accept Dismiss

When the region is created, the subdivisions along the edges will correspond with the existing edge subdivisions so that the mesh will be continuous.

Along the X axis and along the 60 degree line, the number of subdivisions must be set as these sides are the outer boundaries of the model. This may be achieved after the region is created with the modify command or it may be set while the region is being drawn with the element size command.

If the element size command is used when creating a region, the maximum size of the elements along the edges of the region are specified. This is in contrast to setting the number of subdivisions along an edge, but accomplishes the same function.

Draw regions → New region ...

... Polygon → Mouse input → Element size

Element size	=	**1/3**

With the element size set, existing points may be used to draw the region:

Draw regions → New region ...

... Polygon → Mouse input → At old point

and click near the following points.

2.45, 0.0
2.4, 0.3
2.2, 0.2
2.2, 0.5
2.2, 0.8
2.3, 0.8
1.2, 2.1
1.2, 2.2
1.6, 2.0

1.7, 2.2
1.9, 1.9
1.7, 1.8
2.0, 1.6
2.1, 1.4
2.3, 1.4
2.5, 1.2
2.3, 1.2
2.4, 0.8
2.4, 0.6
2.7, 0.6
2.7, 0.3
2.5, 0.3
2.5, 0.0

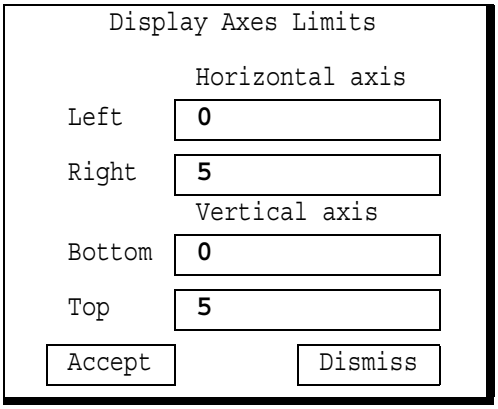
Draw regions → New region ...
... Polygon → Mouse input → Close polygon

Region 8 is created.

Draw regions → New region ...
... Polygon → Mouse input → Return
Draw regions → New region ...
... Polygon → Return
Draw regions → Return

To display the completed model:

MODEL ↓
Zoom display → Numerical axes limits



In addition to creating the geometry, boundary conditions should be assigned to the model and BH curves associated with each magnetic material.

Vector Fields supplies a library of standard BH curves with OPERA-2d. In this problem, two of the standard BH curves (*unisil23m3.bh* and *ndfebo1t.bh*) will be used. The *unisil23m3.bh* curve is a laminate machine steel characteristic that will be assigned to the rotor and stator (material labels 4 and 5). The *ndfebo1t.bh* curve is a typical neodymium iron boron magnet material characteristic and will be assigned to the permanent magnet (material label 6).

Both curves will be loaded from the directory, *$vfdir/bh*[1], and stored in the current working directory. Begin by:

MODEL ↓
BH or
DE data → **BH/DE editing** → **Material 4** → **Load from file**

A file box appears. Change to the directory from which the BH data is to be loaded, i.e.

$vfdir/bh, where *$vfdir* should be replaced by the actual directory in which OPERA-2d was installed[1].

This is done by selecting an item in the sub-directory list and pressing the **CD** button (or double clicking on them).

The file *unisil23m3.bh* should then be selected from the list and **Accept** selected.

The *unisil23m3.bh* curve is displayed.

Now the same BH curve will be assigned to material label 5 by:

BH or DE data → **BH/DE editing** → **Material 4** → **Return**
BH or DE data → **BH/DE editing** → **Material 5** → **Load from file**

and select *unisil23m3.bh* from the file filter box.

The bh curve *unisil23m3.bh* is now assigned to both material label 4 and material label 5. The permanent magnet characteristic should now be assigned to material label 6. Return to the Material list:

BH or DE data → **BH/DE editing** → **Material 5** → **Return**
BH or DE data → **BH/DE editing** → **Material 6** → **Load from file**

and the select the file *ndfebo1t.bh* from the list

BH or DE data → **BH/DE editing** → **Material 6** → **Return**

1. $vfdir refers to the directory in which OPERA was installed. See “Implementation Notes” .

BH or DE data → BH/DE editing → Return
BH or DE data → Return

Boundary Conditions

First redisplay the model:

Zoom display → Same size

The model is displayed on the screen.

A tangential magnetic flux boundary condition has to be assigned to the outer surface of the stator. This means that a fixed value of magnetic vector potential has been assigned here. Physically this means that it is assumed there is no flux leakage from the back of the stator. This is done by

MODEL ↓
Boundary cond. → Vector pot → B normal = 0

and click inside regions 1,2 and 3 near

5.4, 0.5
3.1, 4.2
5.2, 1.2

At the 0 and 60 degree lines of the model, periodic boundary conditions should be set. Do this by:

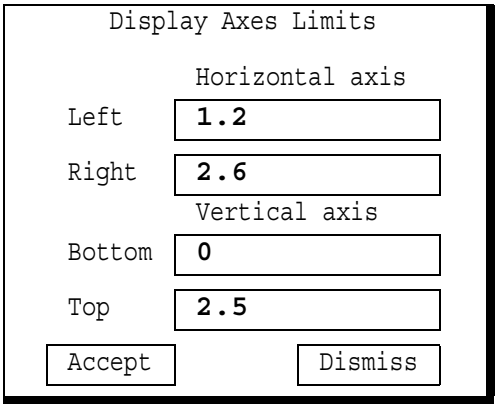
Boundary cond. → Vector pot → Periodic symm.

and then select the points

inside region 1 near (5.0,0.1)
inside region 1 near (4.0,0.1)
inside region 1 near (2.6,0.1)
inside region 6 near (1.6,0.1)
inside region 7 near (0.5,0.1)
inside region 7 near (0.3,0.4)
inside region 6 near (1.0,1.5)
inside region 2 near (1.5,2.2)
inside region 2 near (2.0,3.0)
inside region 2 near (2.7,4.2)

and then zoom in to select some of the smaller sides by

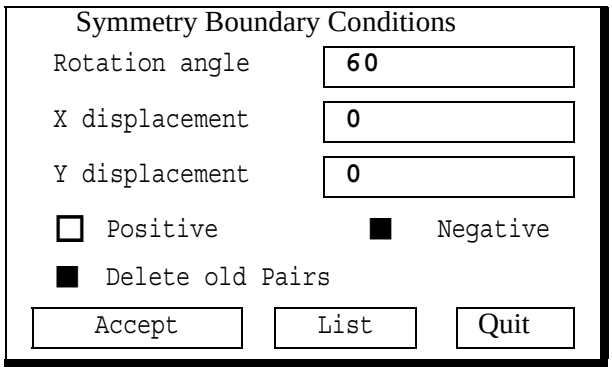
Boundary cond. → **Vector pot** → **Zoom display** → **Numerical axes limits**



Boundary cond. → **Vector pot** → **Periodic symm.**
inside region 8 near (2.47,0.1)
inside region 8 near (1.3,2.1)
Boundary cond. → **Vector pot** → **Return**
Boundary cond. → **Return**

When periodic boundary conditions are specified, the user must tell the program how to match up the pairs of region sides which are to be periodically associated. Do this by:

MODEL ↓
Periodic symm.



A message box opens listing the potential pairs. Hit any key to close the message box.

To generate the finite element mesh:

MODEL ↓
Mesh generator → **generate mesh**

This performs several check functions to ensure that the data to be submitted to the analysis modules is correct.

The mesh is generated. A message box appears reporting progress. The graphics display is a line drawing of the model. The outer boundaries of the model will be drawn in red with the interior model drawn in white.

Any red lines in the interior of a magnetic model indicate the mesh is not continuous. These errors must be corrected before submitting the model for analysis.

The final message box should indicate 0 errors and 0 warnings. Any errors reported must be corrected, while warnings may need to be corrected. Hit any key to clear the screens.

Next, the analysis module which is to be used needs to be specified. From the main menu select

FILE ↓
Write file → **Analysis data**

Select the static analysis module (ST) and accept all defaults from the static analysis options by hitting **Return.**

The model must now be stored for analysis.

Write the model to a file *pmm.op2* using

FILE ↓
Write File → **Write model**

and entering the name **pmm** in the selected file section. Press **Accept** to write the data to file.

This completes the pre processing stage of the example.

Analysis

In order to launch the Static Analysis module select

FILE ↓
Start analysis

Within the **Start Analysis** menu select the appropriate filename (*pmm*) and **Accept**.

OPERA-2d then reports on the progress of the solution.

Upon completion the program returns to the Initial Menu.

Post Processing

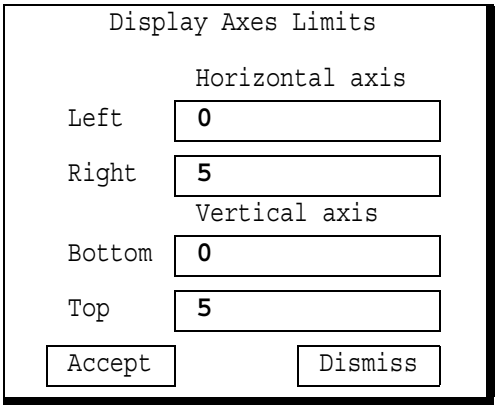
Setting the Pre and Post Processor Environment

Read in the solution file *pmm.st*. A message box is displayed showing the units setting. This can be cleared by hitting any key or mouse button.

Use the **DISPLAY** menu to resize and redraw the screen for viewing the model:

DISPLAY ↓

Axes limits



DISPLAY ↓

Style → **Fill materials**
Style → **Return**
Nodes → **No nodes/vertices**
Nodes → **return**
+Labels ***(toggles to -labels)***
Refresh

In addition to changing the size of the display other changes have been made:-

- **Style** (fill materials)

  This causes the different materials to be outlined (instead of each region).

- **Nodes** (no vertices/nodes)

  This causes no nodes to be displayed.

- **-Labels** (toggles from +Labels)

  This causes no region numbers to be displayed.

Before investing a great deal of time in the analysis of any model, the user should verify that the results being analysed are reasonable.

Line Contours of Vector Potential

One of the easiest things to check is the flux pattern. The static analysis is solved in terms of vector potential and equipotential line contours, in XY symmetry, are equivalent to lines of magnetic flux. To see contours of vector potential click on

FIELDS ↓

Contour plot → **Execute**

A message box displaying the model RMS error is given. This can be cleared by hitting any key or mouse button. The equipotential line contours of vector potential over the whole model are displayed as shown in Figure 9.9. These should be checked to ensure they agree with the expected magnetic performance of the motor. If errors appear to be present then the model should be modified to correct the error and the analysis re-run.

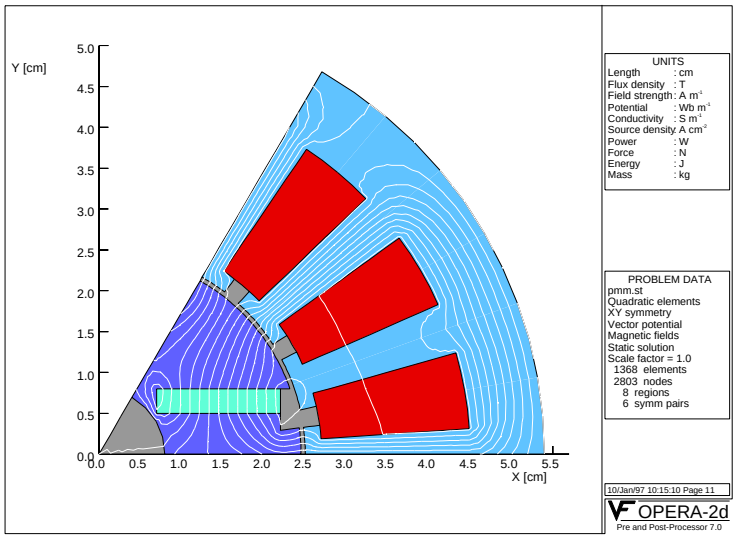


Figure 9.9 Equipotential line contours of vector potential - Equivalent to lines of magnetic flux

Zone Contours of Flux Density Magnitude

In addition to the magnetic lines of flux, the magnetic flux density is a good indication of magnetic performance. The contours may be shown in a number of dif-

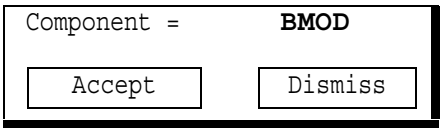
ferent styles, for example filled zones. This displays zones or bands on the model geometry. Each colour band represents a range of values.

To look at flux density one must change the component to be analysed from the default (POT) to BMOD (modulus i.e. magnitude of magnetic flux density **B**) before selecting the contour plot.

Begin by:

FIELDS ↓

Component



then click on

FIELDS ↓

Contour plot → Style → Filled zones
Contour plot → Style → Return
Contour plot → Number of lines

and 50 lines should be specified in the parameter box.

FIELDS ↓

Contour plot → No refresh ***(toggles to Refresh)***
Contour plot → Execute

The magnitude of magnetic flux density is shown in Figure 9.10.

Line Integral

To calculate the total force and torque on a body, in this case the torque on the rotor, an integration surface around the body is defined. Working in two dimensional space, OPERA-2d calculates the total force and torque on a body by performing a series of line integrals encircling the body. For the case of XY symmetry, the calculation is based on a unit length distance in the third direction. The line integrals must be in an anti-clockwise direction and the torque is evaluated with respect to an action point. This gives the x and y force and the anti-clockwise torque around a specified action point (default value is 0,0). For the rotor sector of this model:

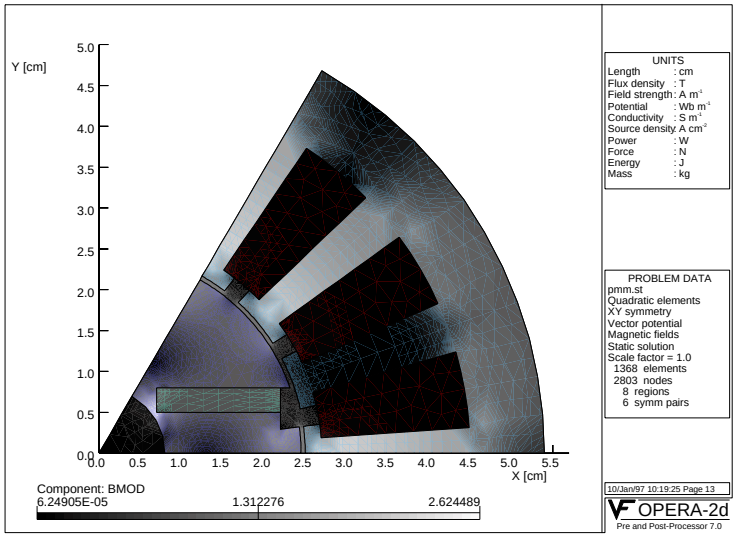


Figure 9.10 Filled zone contours of magnetic flux density magnitude

`Integrals → Along a line`

Start X coordinate	=	**2.47**
......Y coordinate	=	**0**
End X coordinate	=	**1.235**
....Y coordinate	=	**2.139**
Curvature of line	=	**-1/2.47**
Error(<1) or No. of steps(>1)	=	**0.01**

Accept Dismiss

A message box opens showing the results of the line integral. This is cleared by hitting any key.

The torque due to the sector of motor modelled may be found by:-

- The product of the torque given in the message box (system variable TORQUE) and axial length (L) of the motor (since torque is calculated as per unit length) i.e.

$$\text{sector torque} = \text{TORQUE} * L$$

- The total motor torque may be determined from the ratio of the number of poles of the motor to the number of poles in the model (p). The total motor torque is the product of the sector torque and number of motor poles to modelled poles ratio i.e.

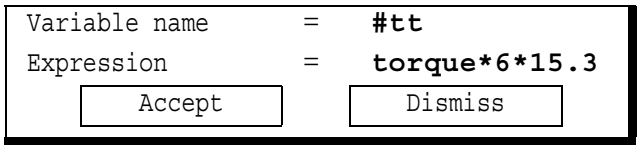
$$\text{Total motor torque} = \text{sector torque} * p = \text{TORQUE} * L * p$$

In this example, the motor has six poles (one was modelled) and is 15.3 cm long. Thus the total torque equation would be

Total motor torque = TORQUE*15.3*6

The built in calculator in OPERA-2d may be used to perform this calculation. The calculator function appears under several of the main menus (**OPTIONS**, **FIELDS**, **MODEL**). Since the **FIELDS** menu is highlighted, begin by:

Calculator



The results will be written into a message box. This can be cleared by selecting the **Continue** button.

The variable name can be up to five characters long, but the first character must be the # symbol. The expression is made up from any FORTRAN 77 style syntax.

This concludes the six pole permanent magnet motor example. To close the graphics window select

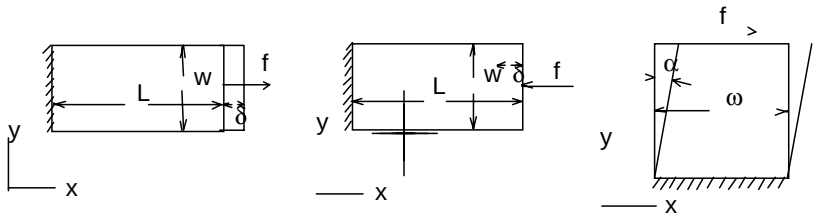
FILE ↓
End OPERA-2d/PP

Chapter 10
Stress Analysis Notes and Examples

A review of stress analysis

Stress

Stress is force per unit area. There are three different forms of stress. In each of the following cases we assume unit thickness normal to the paper.



(a) Tension (b) Compression (c) Shear

Figure 10.1 Definitions of stress

E = Young's modulus of elasticity (unit: force per unit area)
G = shear modulus (unit: force per unit area)

(a) Tension

Tensile stress (force per unit area) $\sigma_x = \frac{f}{w}$

Strain (dimensionless) $\varepsilon_x = \frac{\sigma_x}{E}$

Extension $\delta = L\varepsilon_x$

By convention tensile stress is positive.

(b) Compression

Compressive stress (force per unit area) $\sigma_x = \frac{f}{w}$

Strain (dimensionless) $\varepsilon_x = \frac{\sigma_x}{E}$

Compression $\delta = L\varepsilon_x$

By convention compressive stress is negative.

(c) Shear

Shear stress (force per unit area) $\sigma_{xy} = \frac{f}{w}$

Shear strain (radians) $\alpha = \frac{\sigma_{xy}}{G}$

In a right-handed axis system, positive shear is defined by the right-hand rule, i.e. α is positive anti-clockwise measured from the undisplaced shape. The shear in the diagram is negative.

Note the definition of shear strain which is not that commonly used by physicists (it is twice the latter).

Plane stress and plane strain problems

Plane stress may be seen by considering Figure 10.1part (a) for tension.

If the rectangle represents a thin sheet of isotropic material which is not prevented from contracting in the direction normal to the paper when it is stretched, then a state of plane stress exists.

If the thickness were infinite in the direction normal to the paper, then each plane parallel to the paper would be constrained to remain plane, and a state of plane strain would exist.

Stresses would be developed in the direction normal to the paper, their magnitudes being determined by the Poisson's ratio of the material.

In practice plane stress and plane strain rarely exist in isolation and the situation will be somewhere between the two extremes.

Note:

- *Plane strain* will usually result in higher calculated stresses.
- *Plane stress* will usually result in higher calculated displacements.

Poisson's ratio

Poisson's ratio is defined as the ratio between transverse strain and longitudinal strain when plane stress exists:

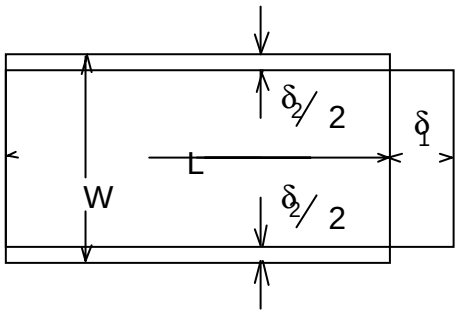


Figure 10.2 Definition of Poisson's ratio

From Figure 10.2, it may be seen that $\varepsilon_1 = \frac{\delta_1}{L}$ and $\varepsilon_2 = \frac{\delta_2}{W}$. This gives Poisson's ratio as $\nu = \frac{\varepsilon_2}{\varepsilon_1}$

Axisymmetry

In axisymmetric problems the stresses are known as

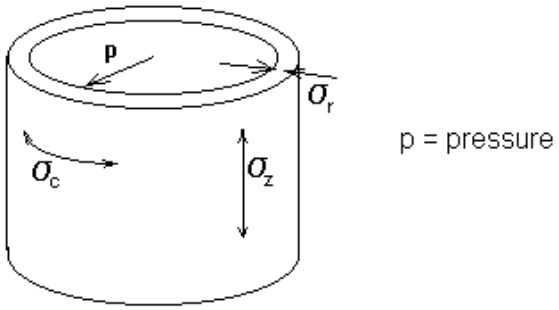


Figure 10.3 The axisymmetric case

- *radial* σ_r
- *axial* σ_z
- *rz shear* σ_{rz}

There is an additional type of stress which is either tensile or compressive:–

- *circumferential stress* σ_c

Assuming the rz plane lies in the surface of the paper, then circumferential stress and strain are normal to the paper.

Circumferential stress is very important in pressure vessels, since it is usually twice the axial stress, and is likely to be the reason for failure (though not always the cause of failure, which might be embrittlement of the material and consequent crack development).

Most magnet problems will be either plane strain or axisymmetric (axisymmetry may be considered as a special case of plane strain).

Finite Element Computation

The Finite Element Method

As for electromagnetic analysis, the structural analysis in OPERA-2d is carried out by discretising the regions of a 2D model into triangular elements.

Each node is said to possess two degrees of freedom. Conventionally these are in the x and y directions. A model with 1000 nodes will therefore have 2000 degrees of freedom (that is, 2000 independent directions in which movement may take place when the model is subjected to stresses and strains.

Corresponding to each degree of freedom, it is possible to formulate an equation which relates the force imposed on that degree of freedom to the set of all the displacements imposed on all the degrees of freedom of the whole model. The set of equations for all the degrees of freedom may be written in matrix form:

$$\mathbf{K}\delta = \mathbf{f} \qquad (10.1)$$

where

K – the global stiffness matrix
δ – the global vector of displacements
f – the global vector of applied forces on the degrees of freedom at the nodes.

The applied forces may be specified, together with any known fixed displacements.

The global stiffness matrix and vector of forces are then formed.

The equation is solved to obtain the global vector of displacements. Using this solution vector, the stresses in the materials are computed. This is known as the *stiffness method*.

Modelling

Stand alone analysis

Using the stress and thermal analysis as stand alone modules, allows the mesh developed to be refined where the gradients of the stress (and displacement) or temperature are expected to be large.

Coupled problems

If electromagnetic results are to be coupled to the stress and thermal analysis modules, it is necessary to transfer information such as Lorentz forces and Joule heating from electromagnetic solution to the structural and/or thermal model. In these cases the mesh developed for the electromagnetic analysis is used for the other two analyses.

The electromagnetic mesh may have been refined in order to obtain accurate estimates of magnetic fields. This may not be adequate in regions where high stresses prevail, such as holes or notches which may not be highly discretised for the electromagnetic solution. A compromise is usually needed.

Saint-Venant's Principle applies in stress analysis:

> "The strains produced in a body by the application, to a small part of its surface, of a system of forces statically equivalent to zero force and zero couple, are of negligible magnitude at distances which are large compared with the linear dimensions of that part of the surface."

This means that the effects of the way in which a force or constraint is applied tend to be localised. It is this principle which allows a force to be applied as a point load, for example, since in theory a point load would inflict an infinite stress at the point of application (in practice it is impossible to apply a force at a point).

Finite Element Mesh

The finite element mesh used are triangles, with linear or quadratic shape functions as for the electromagnetic analysis.

- A linear shape function assumes that, given the nodal displacements of an element, the displacement at any point in the element may be expressed as a linear function of the nodal displacements
- A quadratic shape function uses a quadratic function of the nodal displacements.

Elements with large aspect ratios are to be avoided. The automatic mesh generator attempts to form elements with small aspect ratios.

Aspect ratios greater than 10 should be used with care. Check the solution carefully to see whether it is a realistic result.

The problems caused by high aspect ratios are that the elements may be incapable of correctly representing local changes in strain and stress. Also an ill-conditioned set of equations may result. When solved these will give a completely incorrect set of displacements.

Example

Consider a region of a model when subjected to bending stresses.

This simple beam is being bent by moments applied to its ends, and which is subdivided into a single layer of triangular elements as shown in Figure 10.4.

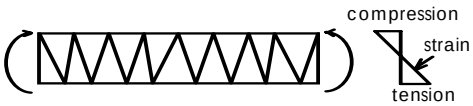


Figure 10.4 A simple beam

The strain in such a beam varies linearly across the depth of the beam, as shown in the right hand diagram of Figure 10.4. A linear triangular element is only capable of representing a constant rate of strain within it (which is why it is often called a constant strain element)

It cannot represent the variation of strain across the beam. Ten or more layers of elements may be needed in such cases.

However, a single layer of quadratic elements would suffice, because the quadratic variation is able to represent linearly varying strain within itself (which is why it is called a linear strain element).

Material Properties

The material type determines how much information must be provided to the pre and post processor.

Isotropic materials

The properties are the same in all 3 orthogonal directions.

- Stress Analysis:

  One value of Young's modulus E and one value of Poisson's Ratio define the material.

  The thermal expansion integral (if needed) will be the same in both directions in the plane

  There is a single value of shear modulus G. Young's modulus, shear stress and Poisson's ratio are related so only two of these need be supplied usually E and ν (the program checks for consistency).

- Thermal Analysis:

  One value of thermal conductivity defines the material.

Anisotropic materials:

Materials may have different properties in three orthogonal directions (the principal axes, set at some angle to the x, y and z axes of the problem).

Such materials are characterised by

E_{ii}	Young's modulus in principal direction *i*
ν_{ij}	Poisson's ratio
G_{ij}	In-plane shear modulus
α_i	Expansion integral in principal direction *i*

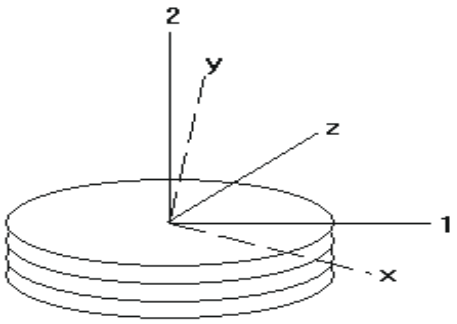
E_{ii} is Young's modulus in a plane defined by the *i* axis.

ν_{ii} is defined as the ratio of strains in a plane *ij* under plane stress (ratio strain in direction *j* to strain in direction *i* for stress in direction i).

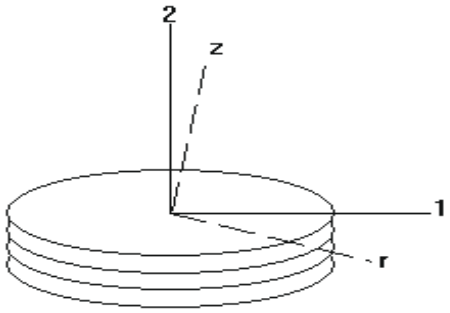
The following types of material may be defined in the pre and post processor.

Orthotropic, Plane stress: specify (E_{11},E_{22},ν_{12},G_{12})
Orthotropic, Plane strain: specify (E_{11},E_{22},E_{33},ν_{21},ν_{31},ν_{32},G_{12})
Transversely isotropic, plane stress: specify (E_{11},E_{22},ν_{12},ν_{21},G)
Transversely isotropic, plane strain: specify (E_{11},E_{22},ν_{12},ν_{21},G)

Axisymetrically stratified: specify (E_{11},E_{22},ν_{12},ν_{21},G)
(in all cases plane 1-2 coincides with plane x-y - see Figure 10.5):



Stata normal to 1-2 and x-y plane
Transversely Isotropic



Strata normal to 1-2 and r-z plane
Axi-Symmetrically Stratified

Figure 10.5 Planar materials

Material properties can vary quite widely depending on the history of the material. When a metal ingot is cast, its Young's modulus and yield strength are usually quite low; they improve dramatically when the material is worked (for example, rolled or drawn). Values quoted in material standards are usually minima for material which has been subjected to normal working.

Constraints

Sufficient constraint must be applied to the model if a valid solution is to be obtained from a structural analysis. In 2D only 3 degrees of freedom are possible for a rigid body: translation in x and y directions, and rotation about the z axis; therefore the minimum number of degrees of freedom requiring constraint is three. However, these must be chosen such that the translations and rotation are

prevented. A 2D model with 1000 nodes will have, after minimal constraint has been applied, 1997 degrees of freedom and will give rise to a set of 1997 simultaneous equations.

It does not matter if a model is over-constrained, providing this is what is requires in the problem. What constitutes sufficient constraint is that translation and rotation of the model as a whole (known as free body movement) must be prevented.

You are not limited to specifying zero displacements at constraints. Specifying non-zero values allow you to compute the effects of initial strains which may be built into your model; such strains might be caused during assembly of the final components, for example.

It is possible to apply skewed constraint: a node, or set of nodes, may be constrained to move along a straight line at a specified angle to the x axis.

Pressures

To specify pressures (which are normal to surfaces), the method used by OPERA-2d is general.

The surface traction is specified. These are the components in x and y (or r and z) directions of forces per unit area of surface. The force may be at any angle (even parallel) to a surface.

The expression analysis capabilities of OPERA-2d make this method much more versatile than is usually the case. For example, to apply pressure to the inside of a hollow sphere, the user defines the LOAD as follows:

r direction: **`p*r/sqrt(r**2+z**2)`**
z direction: **`p*z/sqrt(r**2+z**2)`**

and when the radial displacement is required to be plotted, COMPONENT is set to

`sqrt(dispr**2+dispz**2)`

Thermal Expansion

It is possible to specify a thermal expansion integral for a material. This is the positive value given by:

$$\int_{T_1}^{T_2} \alpha \, dT \qquad (10.2)$$

where α is the expansion coefficient, and the material is heated from T_1 to T_2.

A negative value represents thermal contraction due to cooling. Changes in dimensions and stresses due to constraint will be computed.

For large changes of temperature, material properties such as Young's modulus may vary. In the case of some polymers and elastomers, the thermal expansion coefficient may be affected by stress.

The thermal expansion integral may be used as a means of specifying initial strains.

Failure Theories

There is no one theory which may be used to predict failure of a stressed component. The choice of theory will depend not only on the type of material, but on the way in which it is used, the nature of the loading etc. Materials which are subjected to cyclic stresses (and electromagnets may be in this class) may fail due to fatigue; this type of failure is extremely complex and may require a great deal of information about the material and the stresses, temperatures and even environment to which the component is subjected (for example, the presence of alkali or acid or even just air may affect the stress and/or time at which failure occurs). Particularly in brittle materials (or materials made brittle locally by temperature or fatigue) failure may start at a notch or crack. For steady stresses the common theories are:

- *maximum stress* failure occurs when the maximum stress (usually tensile) in the material reaches a limiting value.
- *maximum strain* failure occurs when the maximum strain reaches a limiting value.
- *maximum shear* failure occurs when the maximum shear stress reaches a limiting value.
- *strain energy* failure occurs when the strain energy reaches the strain energy at failure in a tensile test.
- *shear strain energy* (or Von Mises theory) failure occurs when the shear strain energy reaches a limiting value determined by a tensile test.

The limiting value may sometimes be taken as the stress (or strain energy) when the material fails completely in a tensile test, but it is probably more useful to use the value at yield (which is when the material ceases to be elastic, and some deformation is permanent).

Note that elastic behaviour may be linear, or non-linear; a material such as glass will (under normal conditions of loading) exhibit linear elasticity all the way up to breaking stress, whereas rubber and plastics will be almost entirely non-linear. Metal such as steel remain reasonably linear until the yield point is approached. Since in general material is used at stresses well below yield, it is usually assumed to have linear elasticity.

Whichever stress is used as the limiting stress, it is necessary to apply a safety factor. The actual stress (or strain energy) in the material is not allowed to exceed the limiting value divided by this factor. Typical safety factors are:

- For ductile materials at room temperature 3 to 4 on the breaking stress, 1.5 to 2 on the yield stress
- For brittle materials such as glass, a much larger factor may be used, such as 10 or 20.

The stresses computed by OPERA-2d may not be the maximum. The maximum stress will be the larger of the two principal stresses, which are obtained from the x and y stresses by:

$$p = \frac{\sigma_x + \sigma_y}{2} \pm \sqrt{\frac{\sigma_x^2 - \sigma_y^2}{4} + \rho_{xy}^2} \qquad (10.3)$$

The maximum shear stress is half the difference of these two principal stresses.

In axisymmetric problems the circumferential stress is the third principal stress, and is likely to be greater than that calculated above.

Much of the above assumes (tacitly) that materials fail in tension. This is not always the case, since high compressive stresses lead to high shear stresses which may cause failure. This may be the mode of failure for a short column.

However, a long column or strut will fail due to instability (i.e. applying an axial load to a thin wooden or metal strip, when the load is increased the strip will suddenly bow).

The Nature of Failure

Failure will not occur at some random position in a material. It will begin to occur at some point where the stress is made higher by the presence of a stress concentrator. This may be a sharp re-entrant corner, a hole or perhaps a crack, scratch or imperfection introduced either during manufacture, or during service.

The designer avoids sharp re-entrant corners by introducing fillets into the design. However there may be little control over scratches and surface imperfections. The reason that these imperfections have little effect is that in ductile materials (most metals) a high local stress causes local yielding, and this local yielding relieves the high stress. Because it is tensile stress that causes failure in this way, highly stressed components are often subjected to surface treatments such as polishing, rolling or shot-preening, which by reducing scratches and introducing compressive surface stresses, offset the tensile stresses at imperfections.

Certain common treatments such as metal plating cause surface embrittlement which may cause subsequent failure, and appropriate relieving heat treatment may be essential.

Example 1 - A Simple Beam

This example of a simple beam will illustrate the use of the pre and post processor with the stress analysis module. Methods of building the model, solving and post processing are given including:

- Extra conditions
- Mechanical material properties
- Constraint conditions
- Load conditions
- Tables for pre and post processing
- Display of deformation

The beam is assumed to be infinitely long in the third (z) direction e.g. a shelf or platform, fixed at the y axis and loaded at the far end. The model is made up of a single region (see Figure 10.6).

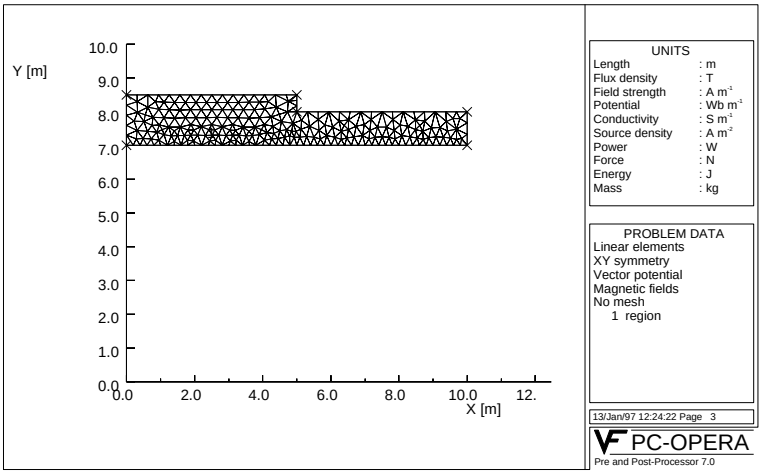


Figure 10.6 The complete model

Pre Processing

Setting the Pre and Post Processor Environment

Launch OPERA-2d and enter the pre and post processor in the normal way. In this example the default S.I. units will be used.

Entering the polygon region

The material definition for the beam is as shown below:

MODEL ↓

Draw regions → **Region defaults ...**

... material type

Material label	=	**3**
Mu or epsilon	=	**1**
Density	=	**0**
Conductivity	=	**0**
Phase/angle	=	**0**
Velocity	=	**0**

Accept Dismiss

The region should have vertices and subdivisions entered using **XY input** with coordinates as follows.

X	Y	Subdivision
0	7	1
10	7	50
10	8	3
5	8	16
5	8.5	3
0	8.5	16

and close the polygon with a subdivision of 5.

This is shown in Figure 10.6

Before adding mechanical data a finite element mesh must be created. Generate the finite element mesh in the normal way, and note that there is a warning relating to potential settings.

Entering the mechanical material data

It is necessary to define the mechanical material data constants for the beam. To do this select

MODEL ↓
Extra options → **Stress/thermal**
materials → **Define a new material**

and complete the parameter box with

Material number	=	3
Accept		Dismiss

Complete the dialogue boxes that follow with the material data:-

Material name = **beam**
Material type = **Isotropic**

and

Accept

Then select **Young's Modulus** from the Stress and Thermal Properties list, and enter a value of **210e9** in the Value box. Select the **Accept** button.

Now select **Poisson's ratio** and enter a value of **0.29** in the value box. Select the **Accept** button and then **Quit**. A message will appear informing the user that material data has been checked successfully.

Entering the mechanical boundary conditions

To constrain the problem select

MODEL ↓
Extra
options → **Extra**
conditions → **Stress/thermal boundaries**

Select the Y component of force of 50 N as follows:–

Load

X or R component	=	0
Y or Z component	=	50
Accept		Dismiss

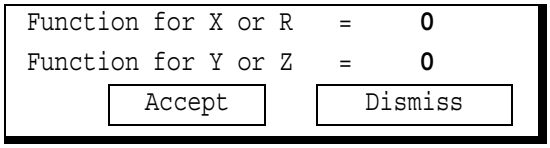
then **Pick sides** and place the cursor at approximately

9.9, 7.75

and close the menu with **Return**

In a similar manner fix the position of the beam at X=0 as follows

Constrained in both



then

Pick sides

and place the cursor at approximately

0.0, 7.75

close the submenus with four **Return** selections.

Completing the Pre Processing

The Stress Analysis data must now be prepared. From the main menu select:

FILE ↓
Write file → **Analysis data**

Click on the Stress Analysis **(SA)** box and subsequently choose the **Plain Strain** option.

The model must be saved. To do this:

FILE ↓
Write file → **Write model**

enter **beam** in the selected file box, and **Accept**. Once the data files have been saved, you can proceed to the next step.

OPERA-2d Stress Analysis Module

A number of analysis modules are supplied with OPERA-2d. The choice of analysis module that depends on the type of problem being solved. (For further information see “Analysis and Utility Programs” on page 5-1). In this particular case, a mechanical stress analysis is required. Hence the stress analysis module is to be used. You may launch the Stress Analysis solver from

FILE ↓
Start Analysis

and **Accept** the highlighted file name for analysis (*beam*).

OPERA-2d then reports on the progress of the solution.

Post Processing

Loading the results into the Pre and Post Processor

Read in the solution file *beam.sa*. You will notice that the solution tables available are also read in automatically. A message box is displayed which informs the user of the new variable names included in the solution file.

Use the **DISPLAY** menu to view the model:

DISPLAY ↓
Style → **Line drawing**

and **Return**

Now select

DISPLAY ↓
Nodes → **No vertices/nodes**

and **Return**

DISPLAY ↓
+Labels *(+Labels changes to -Labels)*
Refresh

Displaying the results as a deformed mesh

The deformation due to loading may be viewed as a deformed mesh.

FIELDS ↓
.....Options *(Below Contour plot)*
Standard plot *(Changes to Deformed plot)*
Deform options

Complete the parameter box by setting the x and y components to **DISPX*1e6** and **DISPY*1e6** respectively (to scale the deformation for viewing). To display the result

FIELDS ↓
Contour plot → **Execute**

The result is shown in Figure 10.7

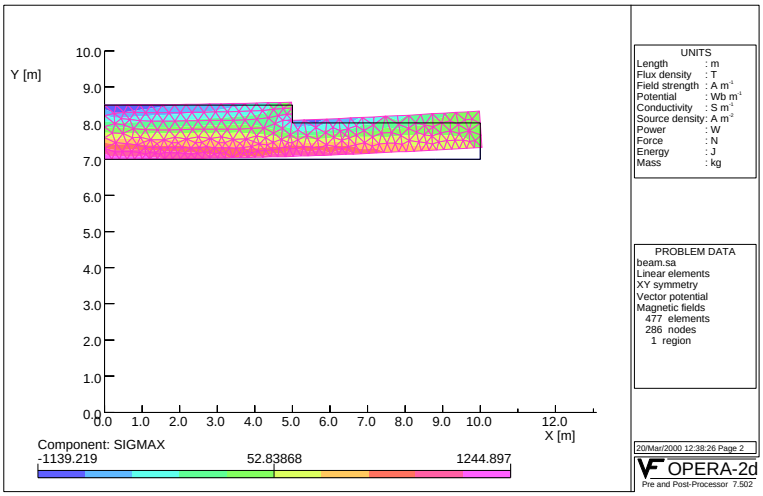


Figure 10.7 Display of Stress Components

Displaying the stress components

To examine the stresses in the beam select:

FIELDS ↓

Component

and enter **sigmax** for the stress component in the x direction. To display the results use

FIELDS ↓

Contour plot → **Style** → **Filled zones**

and

Return

Now select **Execute**

This is shown in Figure 10.8.

Other components may be examined in a similar way with or without the deformed mesh options set.

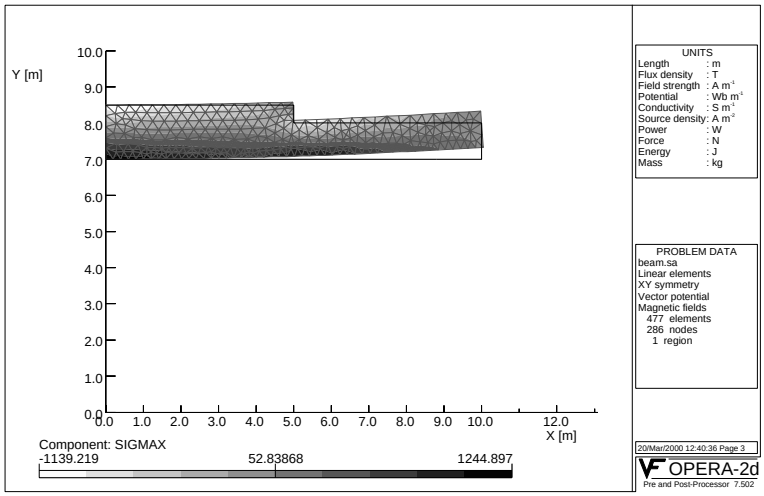


Figure 10.8 Display of stress components

Leaving the Pre and Post Processor

To leave the pre and post processor select **`Return`** followed by:

`FILE` ↓

`End OPERA-2d/PP`

Example 2 - Coupled Problems

This example will illustrate the use of the pre and post processor including the method of coupling the results from the static electromagnetic analysis module to the stress analysis module. Methods of transferring results, solving and post processing are given including:

- Transfer of maxwell stress results using tables
- Extra conditions
- Mechanical material properties
- Constraint conditions
- Load conditions
- Display of deformation

The problem is one of a disc placed above a coil as shown in Figure 10.9

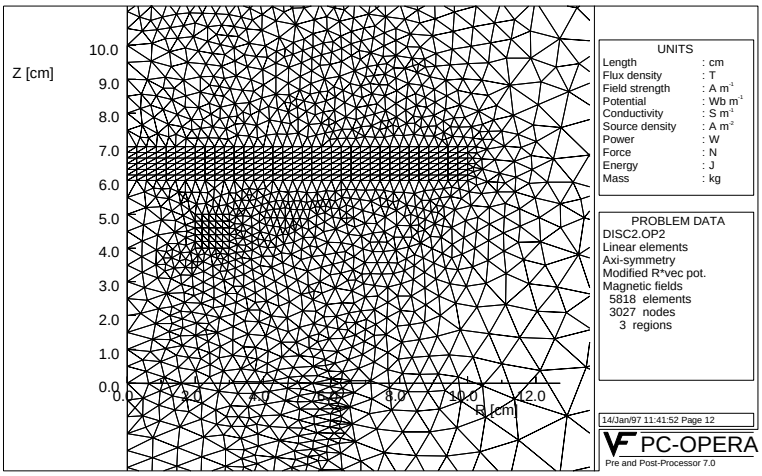


Figure 10.9 The complete model

Initially the electromagnetic problem is solved. The results are read into the pre and post processor and transferred with additional mechanical information to a data file for input into the stress analysis. The results of this analysis are then examined by further post processing.

Solving the Electromagnetic Problem

The electromagnetic model

Launch OPERA-2d and enter the pre and post processor as described in previous tutorials. In this example the default S.I. units will be used. It is assumed that the earlier tutorials have been used and only a brief description of the model is given here.

Set the units to have length expressed in CM. Set the problem to axi-symmetric and the solution potential to modified rA by selecting:

MODEL ↓
Solution type → **Axi symm and potentials**

and from the Axi-symmetry menu select:

MODEL ↓
Solution type → **Axi symm and potentials** → **Modified r * A**

and then **Return**

Define region 1 (the disc) to have:-

- 6 × 35 regular (shape=H) mesh[1]
- Material label = 3
- Permeability, μ = 300

The coordinates of the vertices can be defined using the XY input with settings

X	Y	Subdivision
0	6	1
10	6	35
10	7	6
0	7	35

and close with a subdivision of 6.

Define region 2 (the conductor) to have:-

- 5 × 5 regular (shape=H) mesh
- Material label = 1

1. Change region shape to H using the **Change regions** item under the **MODEL** menu

- Permeability, $\mu = 1$
- Current density = 1e7 A/m^2

The coordinates for the vertices are:-

X	Y	Subdivision
2	5	5
3	5	5
3	4	5
2	4	5

and close with a subdivision of 5.

A background region may now be used to model the surrounding air.

Define this region to have:-

- Background region.
- Material label = 0.
- Permeability, $\mu = 1$.
- Current density = 0.

The coordinates, subdivision and bias for the geometry are:-

X	Y	Subdivision	Bias
0	50		
50	50	9	0.1
50	-50	12	0.5
0	-50	9	0.9

and close with 200 subdivisions and a bias of 0.5

Set the boundary conditions so that all region edges on the Z axis have the **`B NORMAL=0`** condition.

Generate the mesh of the model to allow a display of the mesh of all the regions. This is shown in Figure 10.9.

Obtaining the electromagnetic results

Solve the problem using the non–linear static analysis module and **read the results into the pre and post processor**.

The solution may be examined as in previous examples. Also the solution forms part of the input to the stress analysis module.

Entering the Mechanical Data

It is necessary to define the mechanical material data constants for region 1. To do this

MODEL ↓
Extra options → **Stress/thermal materials** → **Define a new material**

and select **Material number=3**.

Complete the following dialogue boxes with the material data as follows:-

Material name = **steel**
Material type = **Isotropic**

followed by Young's modulus = 210e9 and Poisson's ratio = 0.29

Entering the mechanical boundary conditions

The disc is to be mechanically fixed at the outer edge. Do this by

MODEL ↓
Extra options → **Extra conditions** → **Stress/thermal boundaries** → **Constrained in both**

with: **Function for X or R = 0**
and: **Function for Y or Z = 0.**

Select **Pick sides** and click on the cursor at approximately

9.9, 6.5

It is also necessary to constrain x movement at the y axis. This is achieved by using **Constrain in X or R**. Do this by selecting **Return** and

Constrained in X or R

Specify **Function X or R = 0** and **Pick sides** and click on the cursor at approximately

0.1, 6.5

After setting these boundary conditions, close the submenus with three **Return**

Creating tables for mechanical analysis

The stress analysis module requires data from the electromagnetic solution, relating the loading of the problem to the finite element mesh, to be available as *tables*. These may be automatically created. For the maxwell stress on each element in the r-direction select

MODEL ↓
Extra
options → **Solution**
tables → **Make a new table ...**
... Options → **Maxwell stress X or R**

and **Return**

To form the table select

Make a new table...
... Make table

and complete the parameter box with

Field component	=	**pot**
Table Number 1	=	**1**
Name	=	**rload**
X-derivative	=	
Y-derivative	=	
Unit expression	=	**FORCU/LENGU**3**

Accept Dismiss

Close the submenu with **Return** and in a repeat the procedure for the z direction

Solution tables → **Make a new table ...**
... Options → **Maxwell stress Y or Z**

and **Return**

To form the table select

Make a new table ...
... Make table

and complete the parameter box with

Field component	=	**pot**
Table Number 1	=	**2**
Name	=	**zload**
X-derivative	=	
Y-derivative	=	
Unit expression	=	**FORCU/LENGU**3**

Accept Dismiss

Check the tables have been correctly formed by selecting **List tables**

The program response should be:

```
Table 1: RLOAD
Discontinuous field defined over elements
Units FORCU/LENGU**3
Table 2: ZLOAD
Discontinuous field defined over elements
Units FORCU/LENGU**3
```

Close all the menus with three **Return**

Storing the model and tables

In order to prepare the analysis data select:

FILE ↓
Write file → **Analysis data**

and select the stress analysis (SA) solver. The model must be saved, including the tables which are needed by the stress analysis module. To include the tables, select

FILE ↓
Write file → **write model**

and enter **disc2sa** in the selected file box selecting both **RLOAD** and **ZLOAD** before pressing **Accept**.

OPERA-2d Stress Analysis Module

Launch the solver by selecting

FILE ↓
Start Analysis

and **Accept** the highlighted file name for analysis **(disc2a)**.

OPERA-2d will report on the progress of the solution and, upon completion of the analysis, the user is prompted to view the results file. It is good practice to do so, hence ensuring that no errors occurred during the analysis. Upon examination of the results file the Top Level Menu will be enabled.

Post Processing

Loading the results into the Pre and Post Processor

Read in the solution file *disc2sa.sa* and note that the solution tables available are loaded automatically.

Use the **DISPLAY** menu to view the model:

```
DISPLAY ↓
      Style → Line drawing
      Style → Return

      Nodes → No vertices/nodes
      Nodes → Return

      Material numbers
```

and complete parameter box to indicate: **Not in material=0**

```
DISPLAY ↓
      +Labels(toggles to -Labels)
      Refresh
```

Displaying the results as a deformed mesh

The deformation due to loading may be viewed as a deformed mesh. To do this deselect the air regions which do not form part of the solution

```
FIELDS ↓
      Contour plot...Options
```

then

```
Select material
```

and specify **Not in 0 (air 0).**

followed by

```
Standard plot(toggles to Deformed plot)
Deform options
```

Complete the parameter box by setting the r and z components to **`DISPR*1e10`** and **`DISPZ*1e10`** respectively. To display the result

`FIELDS` ↓

`Contour plot → Label style → No labels`

also select

`No refresh` *(toggles to* **`Refresh`***)*

and

`Execute`

The result is shown in Figure 10.10

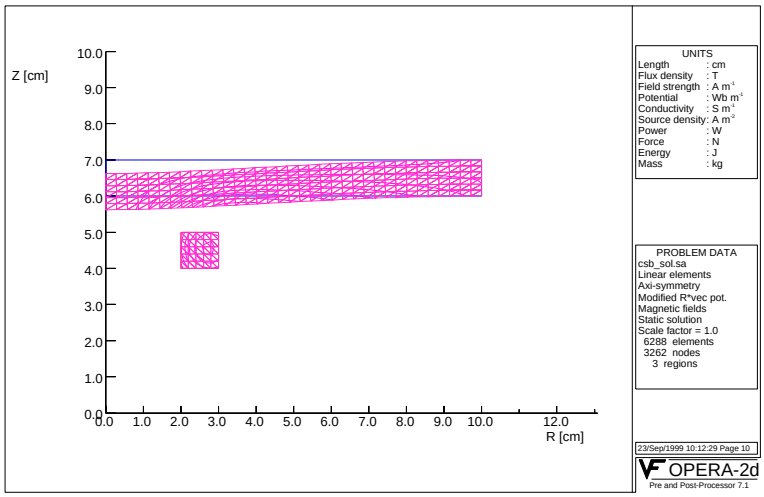


Figure 10.10 Display of the deformed mesh

Displaying the stress components

`FIELDS` ↓

`Component`

and set **`Component = SIGMAZ`** and to display the results using

`FIELDS` ↓

`Contour plot → Style → Filled zones`
`Contour plot → Style → Return`

`Contour plot → Label style → Default labels`

`Contour plot → Execute`

This is shown in Figure 10.11.

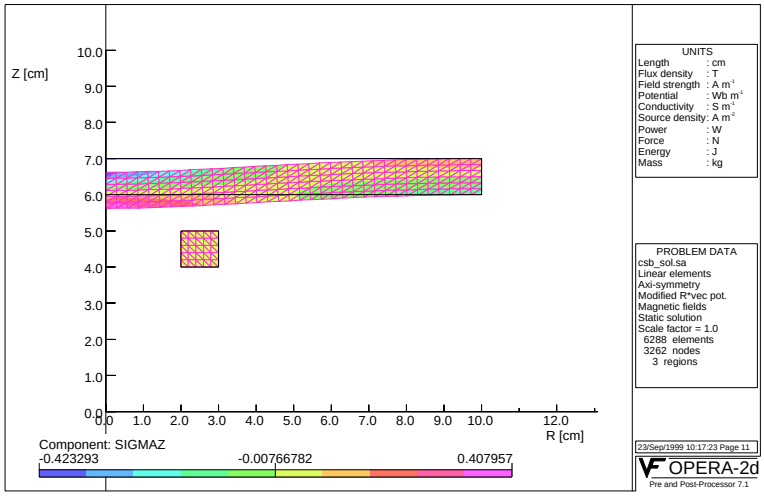


Figure 10.11 Display of stress components

Other components may be examined in a similar way with or without the deformed mesh options set.

Leaving the Pre and Post Processor

Close the menu with **`Return`**

and leave the pre and post processor with

`FILE` ↓

`End OPERA-2D/PP`

Chapter 11
Thermal Analysis Notes and Examples

Thermal Analysis

Performing steady state thermal analysis is similar to electrostatic analysis, with electric potential replaced by temperature. Each node of the finite element mesh possesses only one degree of freedom, which is temperature.

The matrix equation is:

$$\mathbf{K}T = \mathbf{q} \tag{11.1}$$

where

T – the vector of nodal temperatures
q – the vector of applied heat loads
K – a matrix derived from the thermal conduction properties of the model regions.

Thermal conductivity is defined by two values orthogonal to one another i.e. in the x and y directions.

Fixing the temperature at just one node may be all that is necessary (but additional fixed temperatures may be necessary to correctly model the problem).

The simplest boundary condition will be where nodal and/or surface temperatures have been defined. The temperature distribution will be computed within the model.

However, the surface heat transfer may be defined according to the boundary equation

$$k\nabla T.n = q + \alpha(T - T_0) \tag{11.2}$$

where k is thermal conductivity, n is the normal to the surface, q is the surface heat flux (usually watts per square metre), and T_0 is the temperature of a fluid medium in contact with the surface. α is the surface heat transfer coefficient (heat flux per degree of temperature difference between surface and fluid). This coefficient may be estimated by using heat transfer correlations which can be found in a text book on heat transfer. Unfortunately, heat transfer coefficients are very dependent on the nature of the surface and the dimensions of the surface and the object, as well as on the fluid. Either q or α may be specified zero.

Definition of surface heat transfer may provide adequate constraint. This would be the case where, for example, Joule heat was being generated in regions of a magnet, and the magnet was being cooled by air, water or other fluid.

If you do not specify the temperature of a surface or the heat transfer across it, then the program will assume

$$\frac{dT}{\mathrm{dn}} = 0 \qquad (11.3)$$

that is, the surface is perfectly insulated. Since in practice there is no perfect insulating material, you may wish to include the effects of imperfect insulation. This can be done either by explicitly modelling layers of insulating material, or by using equivalent values of heat transfer coefficient.

Example 1 - A Simple Heat Bath

This example of a simple Heat Bath will illustrate the use of the pre and post processor with the thermal analysis module. Methods of building the model, solving and post processing are given including:

- Extra conditions
- Thermal material properties
- Fixed temperature conditions
- Perfect insulation conditions
- Thermal flow conditions
- Tables for pre and post processing
- Display of temperature distribution

The bath is assumed to be infinitely long in the third (z) direction with a fixed temperature at the right hand end. The model is made up of two regions as shown in Figure 11.1.

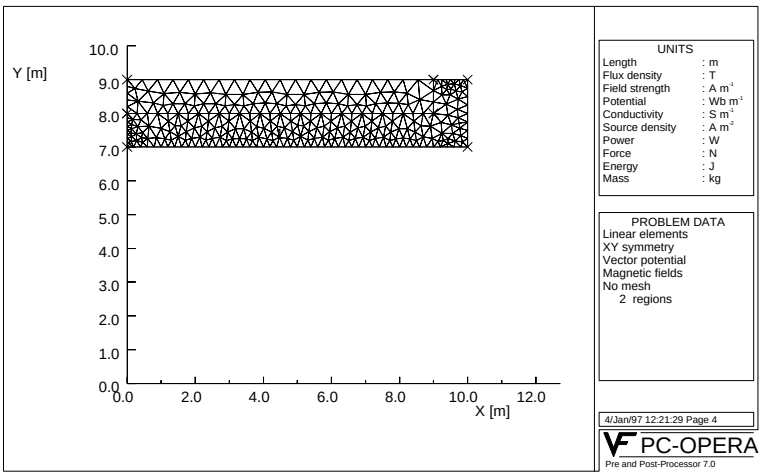


Figure 11.1 The complete model

Pre Processing

Setting the Pre and Post Processor Environment

Launch the pre and post processor in the normal way. In this example the default S.I. units will be used.

Entering the polygon region

The general definition of the bath is shown below for the two regions of the model. Region 1 is defined by:

MODEL ↓
Draw regions → Region defaults ...
... material type

and set

Material label	=	3
Mu or epsilon	=	1
Density	=	0
Conductivity	=	0
Phase/angle	=	0
Velocity	=	0

Accept Dismiss

The region should have vertices and subdivisions as follows

(0,7) to (10,7) with 50 subdivisions.
(10,7) to (10,9) with 8 subdivisions.
(10,9) to (9,9) with 8 subdivisions.
(9,9) to (9,8) with 3 subdivisions.
(9,8) to (0,8) with 30 subdivisions.
(0,8) to (0,7) with 8 subdivisions

and region 2 is defined by:

MODEL ↓
Draw regions → Region defaults ...
... material type

and set

Material label	=	5
Mu or epsilon	=	1
Density	=	0
Conductivity	=	0
Phase/angle	=	0
Velocity	=	0

Accept Dismiss

The region should have vertices and subdivisions to match region 1. In addition:

(0,8) to (0,9) with 5 subdivisions.
(0,9) to (9,9) with 20 subdivisions.

This is shown in Figure 11.1

Before adding thermal data a finite element mesh must be created. Generate the finite element mesh in the normal way, and note that there are warnings relating to potential settings and BH data.

Entering the thermal material data

It is necessary to define the thermal material data constants for the bath. To do this

MODEL ↓
Extra options → **Stress/thermal**
materials → **Define a new material**

and select

Material number	=	3

Accept Dismiss

Complete the following dialogue boxes with the material data as follows:-

Material name = **bath**

and select **Thermal** from the Material Type Options list.

then

Thermal conductivity with values of **63** (W/m/K) for each of the x and y components, the value of each component being separated by a space.

Repeat the procedure for the second material (number = 5) of region 2 with:

Material name = **melt**

and select **Thermal**.

then

Thermal conductivity with a value of **400** (W/m/K) for each of the x and y components (separate the two identical values by a space).

Entering the thermal boundary conditions

MODEL ↓
Extra
options → **Extra**
conditions → **Stress/thermal**
boundaries → **Fixed temperature**

and set Temperature = **500** (^{o}C).

Select **Pick sides** and click the mouse at approximately

5.0, 7.1
9.9, 8.0

Close the submenu with **Return**

Select **Heat transfer** and set

Heat flux = **50**,

Heat transfer coefficient = **1**

Coolant temperature = **120**

Select **Pick sides** and click the mouse at approximately

5.0, 8.9

Close the submenu with **Return** and select **Heat transfer** and set

Heat flux = **0**

Heat transfer coefficient = **0.01**

Coolant temperature = **150**

Select **Pick sides** and click the mouse at approximately

9.5, 8.9

Close the submenus.

Completing the Pre Processing

In order to prepare the analysis data select

FILE ↓
Write file → **Analysis data**

Select the Thermal Analysis module **(TH)** and **Return.**

It is now necessary to store the model by selecting

FILE ↓
Write file → **Write model**

entering **bath** as the selected file name and pressing **Accept**.

Thermal Analysis Module

To run the thermal analysis select:

FILE ↓
Start Analysis

from the menu and then **Accept** the highlighted filename **(bath).**

OPERA-2d will report on the progress of the solution as it proceeds.

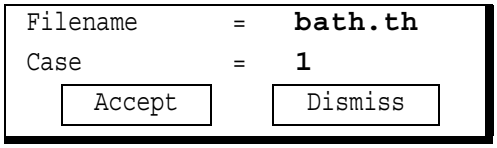
Examining the results

Loading the results into the Pre and Post Processor

In order to post process the problem, read in the solution file *bath.th*

FILE ↓
Read file → **Read model**

and enter



A message box is displayed informing the user that table **TEMP** has been loaded. This may be cleared by hitting any key or mouse button.

Use the **DISPLAY** menu to view the model:

DISPLAY ↓
Refresh

To view the temperature distribution, set the component to be **TEMP** and use a filled zone contour plot.

FIELDS ↓
Component=temp
Contour plot → Style → Filled zones

then

Execute

The result is shown in Figure 11.2.

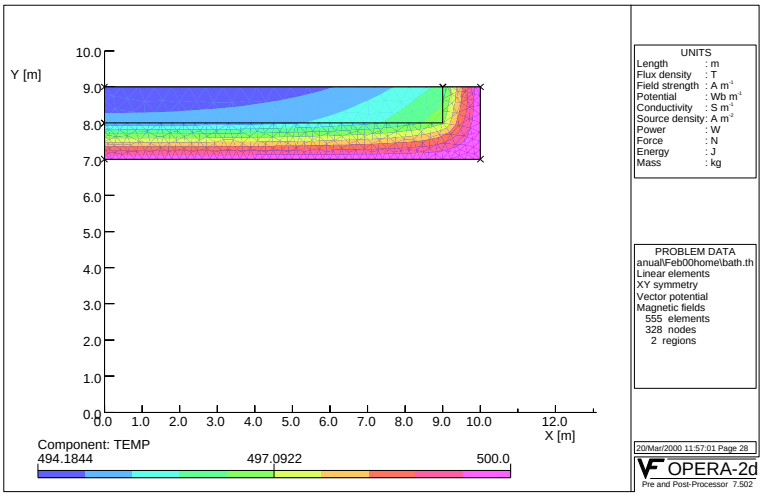


Figure 11.2 Displaying the temperature distribution

Leaving the Pre and Post Processor

Close the submenu with **Return**

FILE ↓
End OPERA-2d/PP

Example 2 - Thermally Coupled Problems

This example will illustrate the use of the pre and post processor including the method of coupling the results from the static electromagnetic analysis module to the thermal analysis module. Methods of transferring results, solving and post processing are given including:

- Transfer of heat results using tables
- Extra conditions
- Thermal material properties
- Temperature constraint conditions
- Heat flow and convection boundary conditions
- Display of temperature

The problem is that of a disc placed above a coil. The model is the same as the one used in the OPERA-2d/SA example (see “Stress Analysis Notes and Examples” on page 10-1), with an additional region. This is shown in Figure 11.3.

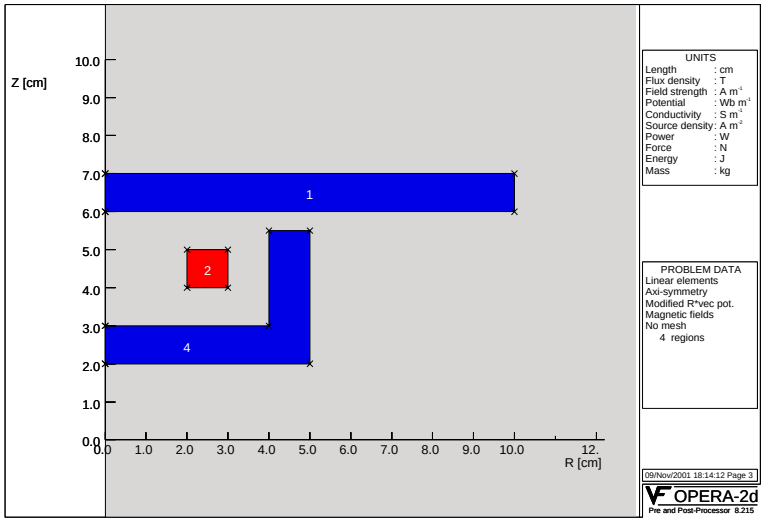


Figure 11.3 The complete model

Initially the electromagnetic problem is solved using the AC analysis module. The results are read into the pre and post processor and transferred with additional thermal information to a data file for input into the thermal analysis. The results of this analysis are then examined by further post processing.

Solving the Electromagnetic Problem

The electromagnetic model

Launch OPERA-2d and enter the pre and post processor in the normal way. The default SI units will be used. Enter the model data as follows.

Set the units to have length expressed in CM. Set the problem type and solution potential by selecting:

MODEL ↓

Solution type → **Axi symm and potentials** → **Modified r*A**

and then **Return**

Define region 1 (the disc) to have:-

6×35 regular (shape=H) mesh (subdivision of 35 to be along the radius of the disc)

Material label = 3

Permeability = 300

The coordinates for the vertices are:-

0, 6
10, 6
10, 7
0, 7

Define region 2 (the conductor) to have:-

5×5 regular (shape=H) mesh

Material label = 1

Permeability = 1

Current density = 1e7 A/m^2

The coordinates for the vertices are:-

2, 5
3, 5
3, 4
2, 4

A background region may now be used to create a far field boundary.

Define this region to have:-

Background region.

Material label = 0

Permeability = 1.0

Current density = 0.0

The coordinates and subdivision number for the geometry are:-

(0,50) to (50,50) with 9 subdivisions and biasing towards the Z axis.
(50,50) to (50,-50) with 12 subdivisions with no bias.
(50,-50) to (0,-50) with 9 subdivisions and biasing towards the Z axis.
(0,-50) to (0,50) with 200 subdivisions and no biasing.

Modify region 1 to have a conductivity by selecting:

MODEL ↓
Modify regions → **Modify region** → **Pick Region**

Now select **Material data** and set Conductivity = **1e6**. Select **Return** three times to close the submenus.

Modify region 2 to have a smaller current density by selecting:

MODEL ↓
Modify regions → **Modify region** → **Pick Region**

Now select **Material data** and set Density = **1e5**. Select **Return** three times to close the submenus.

Add an additional region. Define this region to have:-

Polygon region.

Material label = 3.

Permeability = 1.

Density = 0

Conductivity = 5e6.

The coordinates and subdivision number for the geometry are:-

(0,3) to (4,3) with 8 subdivisions.
(4,3) to (4,5.5) with 8 subdivisions.
(4,5.5) to (5,5.5) with 5 subdivisions.
(5,5.5) to (5,2) with 10 subdivisions.
(5,2) to (0,2) with 10 subdivisions.
(0,2) to (0,3) with 5 subdivisions.

Generate the mesh of the model to allow a display of the mesh of all the regions. This is shown in Figure 11.4. Set the boundary conditions such that all region edges of the Z axis have the **`Bnormal=0`** condition. (Note this boundary condition must be added to the new region if the old model has been read in).

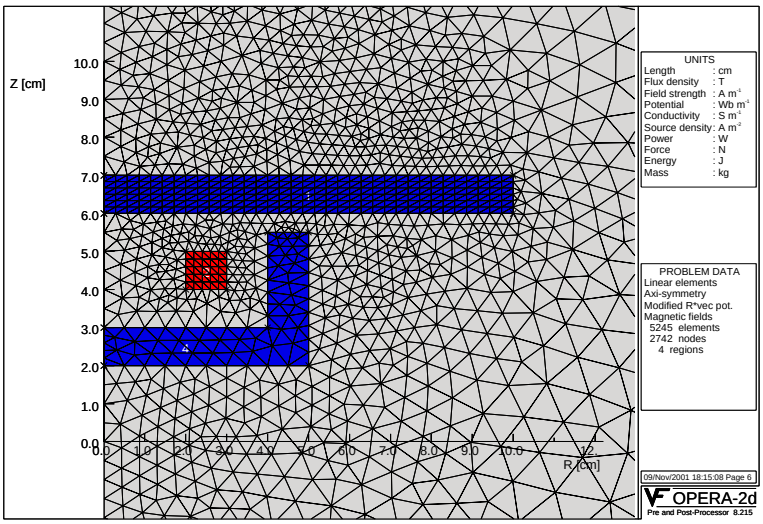


Figure 11.4 The complete model including mesh

Obtaining the electromagnetic results

Write the solution to a file called (say) *disc50.op2*, using the default frequency (50 Hz). Solve the problem using the linear a.c. analysis module and read the results into the pre and post processor.

The solution may be examined as in previous examples. Also the solution forms part of the input to the thermal analysis module.

Entering the Thermal Data

It is necessary to define the thermal material data constants for region 1 and 4. To do this

`MODEL` ↓
`Extra`
`options` → **`Stress/thermal`**
`materials` → **`Define a new material`**

and select **`Material number=3`**.

Complete the following dialogue boxes with the material data as follows:-

Material name = **steel**

and select **Thermal**

followed by:

Thermal conductivity = **100 100**

Entering the thermal boundary conditions

The disc has no fixed temperatures at the outer edges. Only convection boundary conditions are to be assigned to the edges. Do this by

MODEL ↓
Extra
options → **Extra**
conditions → **Stress/thermal**
boundaries → **Heat transfer**

with

heat flux = **0**

heat transfer coefficient = **0.1**

coolant temperature = **25.0**

Select

Pick sides

and click on the cursor at approximately

5.0, 6.9
5.0, 6.1
9.9, 6.5
4.5, 5.4
4.1, 4.0
4.9, 4.0
2.0, 2.9
2.0, 2.1

The boundaries along the z axis may be represented by perfect insulation and so do not require setting, as this is the default condition.

Creating tables for thermal analysis

The thermal analysis module requires data, from the electromagnetic solution, relating the heat sources of the problem to the finite element mesh, to be available as *tables*. These may be automatically created. For the heat generated on each element select:

Field ↓
Solution tables → **Make a new table ...**
... options → **Select material**

and specify

Only in material	=	**3**
Not in material	=	**0**
Accept		Dismiss

and **Return**

To form the table select

MODEL ↓
Extra options → **Solution tables** → **Make table**

and specify:

Field component	=	**J**2/SIGMA**
Table number	=	**1**
Name (1 to 10)	=	**HEAT**
X Derivative name		
Y Derivative name		
Unit expression	=	**POWEU/LENGU**3**
Accept		Dismiss

then select **ACCEPT** and close the submenu with **Return**

Check the table has been correctly formed by selecting **List tables**

The program response should be:

```
Table 1: HEAT
Discontinuous complex field defined over elements
Units POWEU/LENGU**3
```

Close all the menus with three **Return**

Storing the model and table

In order to prepare the analysis data select

FILE ↓
Write file → **Analysis data**

Select the Thermal Analysis module **(TH)** and **Return.**

The model must be saved to include the table HEAT which will be used by the thermal solver. To do this select

FILE ↓
Write file → **Write model**

Select table HEAT and complete the selected file box with **disc2th** and **Accept**. Once the data files have been saved the Analysis Module can be launched.

OPERA-2d Thermal Analysis Module

A number of analysis modules are supplied with OPERA-2d. The choice of analysis module that depends on the type of problem being solved. In this particular case, a thermal analysis is required. Hence the thermal analysis module is to be used. Select

FILE ↓
Start Analysis

and select the highlighted *disc2th.op2* file.

On completion the program returns to the Top level Menu.

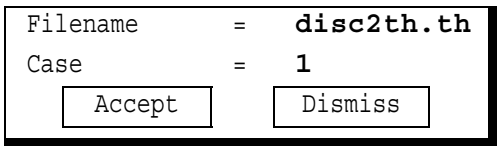
Post Processing

Loading the Results into the Pre and Post Processor

Read in the solution file *disc2th.th* together with the solution tables available. To do this

FILE ↓
Read file → **Read model**

and complete the parameter box with



(If the *disc2th.mesh* file has been deleted, it is necessary to remesh the *disc2th.th* file before continuing). A message box is displayed and this may be cleared by hitting any key or mouse button.

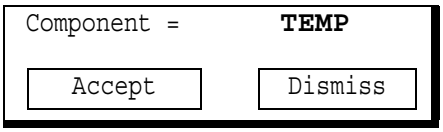
Use the **DISPLAY** menu to view the model:

DISPLAY ↓
Refresh

Displaying the temperature distribution

FIELDS ↓
Component

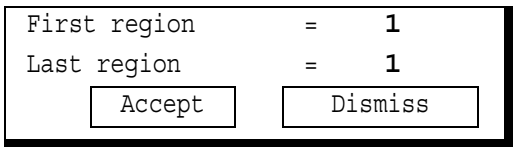
and set



and display the results using

FIELDS ↓
.....Options (Below Contour Plot)

and select **Select regions** with



Close the submenu with **Return** and select

Contour plot → Style → Filled zones
Return
Execute

The result is shown in Figure 11.5.

Repeat the procedure for region 4 using:

Contour plot → Refresh

The results are shown in Figure 11.6.

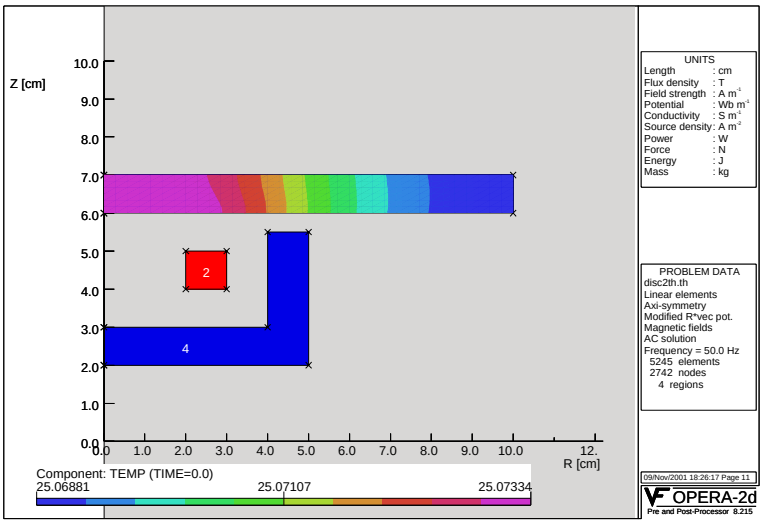


Figure 11.5 Display of temperature in region 1

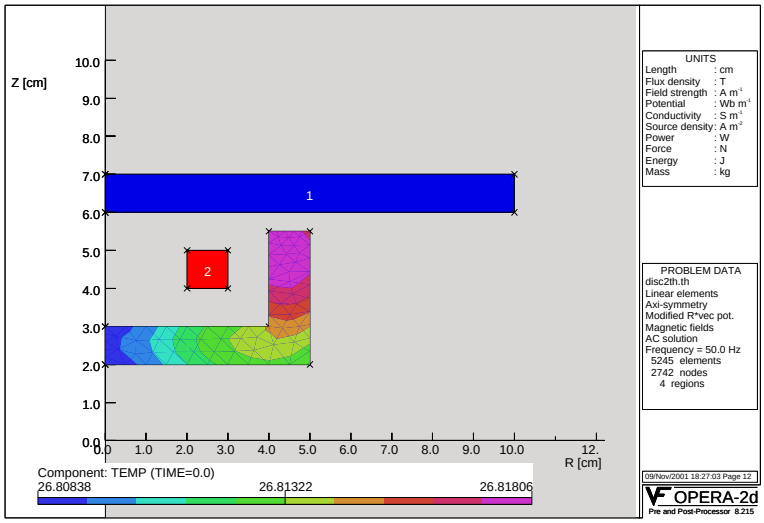


Figure 11.6 Display of temperature in region 4

Leaving the Pre and Post Processor

Close the menu with **`Return`**

and leave the pre and post processor with

`FILE` ↓

`End OPERA-2D/PP`

and confirm.

Chapter 12
Space Charge Example

Introduction

Space Charge is an effect created by dense beams of charged particles. A high current beam from an electron gun exhibits the space charge effect. This example shows the space charge effect produced by a typical electron gun.

Space Charge models can include the effects of magnetostatic and electrostatic fields or electrostatic fields alone. This example illustrates electrostatic fields only with potentials assigned on the electrodes. Thus only air regions are required (with an additional conductor region being added to terminate the beam). The complete model is shown in Figure 12.1.

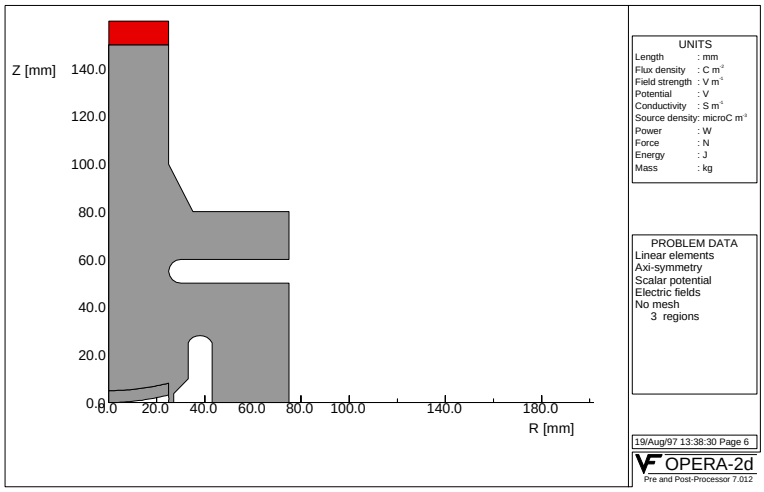


Figure 12.1 The complete space charge model

Features included in this tutorial are:

- Emitter definitions and files
- Shape H regions
- Space charge analysis module

OPERA-2d Pre Processor

Setting the Pre Processor Environment

Launch the pre and post processor in the normal way. The model will be drawn with the following settings:

- axi-symmetric coordinates
- Solution type of ELECTRIC field
- SCALAR potential
- SI electric units with lengths in mm

This is carried out as follows:

MODEL ↓

Solution type → **Axi symm and potentials** → **Scalar Potential**

Solution type → **Axi symm and potentials** → **Return**

Solution type → **Electric Field**

A warning message is given stating that the units of magnetic field quantities are not valid and that units will be set to take account of the solution potential change.

The units are set by

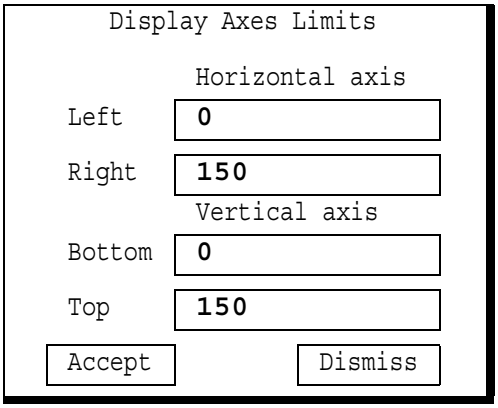
UNITS ↓

SI Units (electric)
Length Unit → **millimetre**

and resize the display by

DISPLAY ↓

Axes limits



Refresh

Building the model

The model is created out of three regions:

- The region in front of the emitter (material air)
- The vacuum tube (material air)
- The target (material conductor)

Region 1

A separate region is used near the emitter surface so that the mesh can be well structured to fit the beam pattern leaving the emitter. This is very important for the convergence of the solution and the accuracy of the results obtained. The default material definition is accepted for region 1 since this is a vacuum. The region coordinates, subdivisions and curvature are listed in the table below. XY coordinate input is suggested in this case.

MODEL ↓

Draw regions → **New region ...**

... Polygon → **XY Input**

X coordinate	=	**0**
Y coordinate	=	**0**
Line curvature	=	**0**
Subdivision	=	**1**
Bias	=	**0.5**

Accept Dismiss

Continue the XY input for the following coordinates that make up the region in front of the emitter.

X	Y	Curvature	Subdivision
0	5	0	6
25	8.17542	-1/100	20
25	3.17542	0	6

Close the region with

Draw regions → **New region ...**

... Polygon → **Close polygon**

Subdivision	=	**20**
Line curvature	=	**1/100**
Bias	=	**0.5**

Accept Dismiss

Region 1 is created. Region 2 is also a vacuum region. It is created in a similar manner, with the following coordinates. (The first 3 points listed in the table below could be selected using mouse and the **At old point** option.)

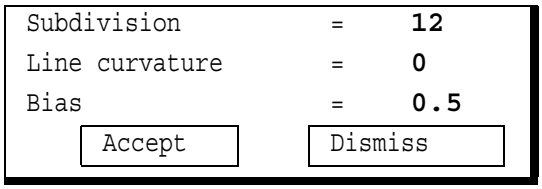
X	Y	Curvature	Subdivision
0	5	0	20
25	8.17542	-1/100	20
25	3.17542	0	6
25	0	0	4
27	0	0	2
27	3.71397	0	6
33	10	0	8
33	25	0	8

X	Y	Curvature	Subdivision
43	25	0.178885	8
43	0	0	8
75	0	0	5
75	50	0	8
30	50	0	8
30	60	0.2	8
75	60	0	8
75	80	0	4
35	80	0	8
25	100	0	8
25	150	0	15
0	150	0	10
0	100	0	15
0	60	0	20
0	50	0	5
0	20	0	15

Close the region with

Draw regions → New region ...

... Polygon → Close polygon



The final region is a conductor region (material 1). It is a feature of the analysis module that as a particle trajectory enters a material 1 region (conductor region), the trajectory terminates and this makes analysis more efficient. The Material type is set prior to drawing the final region as follows:

MODEL ↓

Draw regions → **Region defaults ...**

... material type

Material label	=	**1**
Mu or epsilon	=	**1**
Density	=	**0**
Conductivity	=	**0**
Phase/angle	=	**0**
Velocity	=	**0**

Accept Dismiss

The region coordinates can again be entered using XY input with the following values.

X	Y	Curvature	Subdivision
0	150	0	1
0	160	0	1
25	160	0	10
25	150	0	1

and the polygon should be closed with

Draw regions → **New region ...**

... Polygon → **Close polygon**

Subdivision	=	**10**
Line curvature	=	**0**
Bias	=	**0.5**

Accept Dismiss

To view the geometry, display the model with the mesh. Do this using:

DISPLAY ↓

-Mesh (***toggles to +Mesh***)

Refresh

Note in particular the triangular mesh in regions 1 and 3. Though the basic input of the model is now complete, several changes and additions need to be made. These include changing the shape code of regions 1 and 3 to give a regular mesh and adding the boundary conditions (i.e. potentials) to the model.

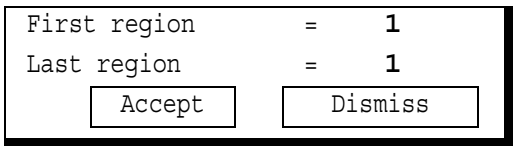
Regions created from the menus are polygons by default. Polygon regions can have many sides and are appropriate in most cases. However four sided polygon

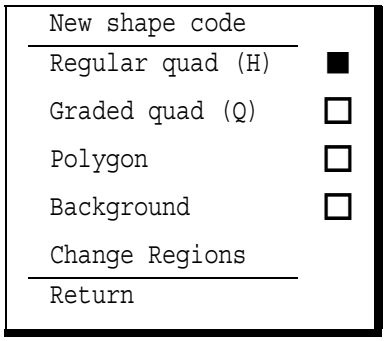
regions can be converted from polygon shape regions to H or Q shape regions. The advantage of H or Q shape regions is that they have a regular mesh.

Regions 1 and 3 are to be converted to shape H regions. This provides a mesh in region 1 able to model the space charge distribution more accurately. To convert the regions:

MODEL ↓

Change Region → **Region numbers**





selecting **Regular quad (H)** followed by **Change regions**.

The same procedure should be applied to region 3, although the mesh in this region is unimportant as the beam will terminate as it enters any region that has material label greater than 0.

Refresh the display to see the effects of these changes. The mesh will now appear as a regular grid in Regions 1 and 3. Use the

DISPLAY ↓

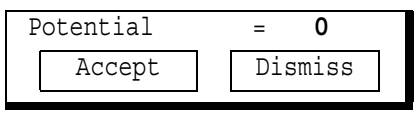
Zoom in/out

menu item to get a clearer view of the mesh in these regions if necessary.

Once this has been completed, set the Boundary Conditions (Potentials) of the model using the following procedure:

MODEL ↓

Boundary Cond. → **Scalar pot** → **Set Potential**



Boundary Cond → **Scalar pot** → **Set Potential** → **Apply potential**

then click near the edge of region 1, close to the coordinates (12,1).

This assigns 0 potential to this edge. The following potentials also need to be assigned. To change the value of potential being assigned select **Return** and set the potential to a new value. It may also be necessary to zoom in on some of the smaller sides so that they can be selected more easily. Figure 12.2 shows the model with the boundary conditions applied.

X	Y	Potential
25.1	1	0
26.9	1	0
30	8	0
32	20	0
38	30	0
44	15	0
55	49	50000
24	55	50000
55	61	50000
55	79	100000
30	90	100000
24	120	100000

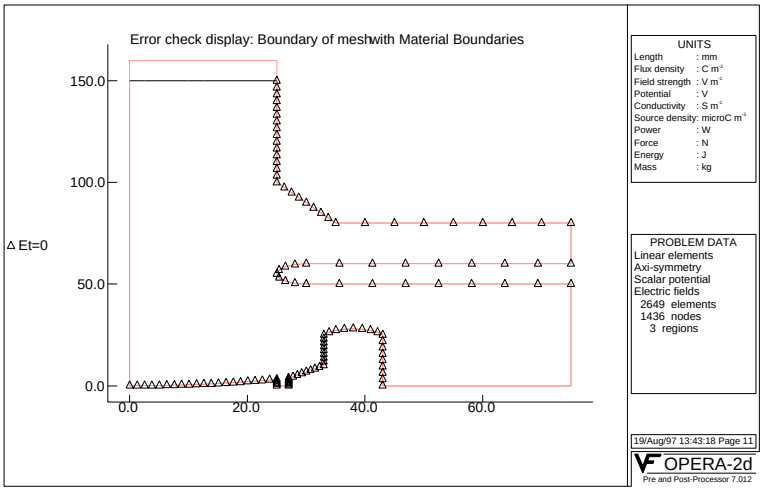


Figure 12.2 Boundary conditions applied to the model

The final step before saving the model is generating the mesh. This is done with the following sequence:

MODEL ↓
Mesh generator → **Generate mesh**

After the program finishes meshing the model, the outline of the model is drawn on the screen, as shown in Figure 12.2. A text box appears which displays the results of the mesh generation. The user should scroll to the bottom of the text box and verify that there are no errors and/or warnings. Any errors or warnings will need to be corrected before running the model.

The analysis data may be set up by selecting

File ↓
Write file → **Analysis data**

and selecting the space charge solver, with all default settings.

The model now needs to be saved. Save as **space.op2** using the following sequence:

File ↓
Write file → **Write model**

and enter **space** as the file name in the File Box.

The space charge example model has now been created.

Creating the Emitter Data

Before running the analysis module, the user needs to provide the program with information which describes the emitter. This is accomplished by defining the emitter parameters in the preprocessor using the model geometry. The definition must be saved as *filename.emit*, where the filename is the same as the model. (i.e. for file *space.op2*, the file must be called *space.emit*)

The emitter file is created using the following sequence:

Model ↓
Emitter data → **Create new emitter**

A message box appears stating

`Number of emitters is 1`

Pick the emitter line segments using

Model ↓
Emitter Data → **Pick region side**

and select the emitter defined, select **Add** followed by **Select Side** and pick the line near the coordinates (12,1) and select **Accept**.

A message box appears stating:

Line segment created in emitter 1 Number of line segment is 1

Arrows are drawn on the screen showing where the particles are to be emitted. Notice that the arrows are pointing in the negative Z direction which is incorrect for this model. To change the direction so that the arrows will point in the positive Z direction:

Emitter Data → **Edit emitter data**

and select **Emitter 1** and **Accept**. Then select:

Emitter Data → **Edit emitter data** → **Edit Line segment data**

and select **Segment 1** and **Accept**. Pressing the **Toggle** button at the bottom of the parameter box changes the direction of the particles, and hence the arrows shown on the display.

Next the emitter data is modified:

Emitter Data → **Edit emitter data** → (**select Emitter 1**)
Edit emitter data

and select **1d Lang/Fry (virt. cathode)** and fill in the parameter box as shown:

Parameter	Value
Emitter temp (Kelvin)	1273
Work func. (Volts	1.82
Emit const. (A/cm2)	350
Particle mass	1
Particle charge	-1
Max. dist. between rays	2

and **Accept**.

Finally edit the emitter global parameters by selecting **Global parameters** and fill in the parameter box as shown:

Parameter	Value
Max. traj. step length	2
Absolute traj. tolerance	0.01
Normal sampling distance	0.1

The emitter data has now been created and must be saved. To do this select **Return**, followed by **Store an emit file**. Fill in the file box using the filename **space**.

Details of the emitter file and different options available are given in the Space Charge section of the Reference Manual. The accuracy of the solution is dependent on the mesh and the emitter matching correctly. Calculation of the currents generated in each beam is dependent upon the space charge calculated near the surface of the emitter. Hence the SHAPE H region was used to generate a uniform mesh over this area with the number of elements matching the number of beams being emitted from the emitter surface.

Now that the *space.op2* and the *space.emit* files exist, the model can be solved with the Space Charge analysis module.

Space Charge Analysis

The model is solved using the Space Charge analysis module, **SP**.

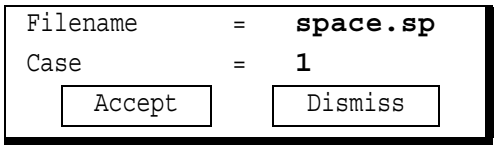
The user should enter the filename **space** when prompted and **Accept** the defaults for the other prompts.

Post Processing

Read the model into the post processor. The file to be read in is *space.sp*. In addition to the solution file, the user should note that the table RHO containing the space charge distribution is automatically read in. (If the mesh has not been saved, it will be necessary to remesh the model.)

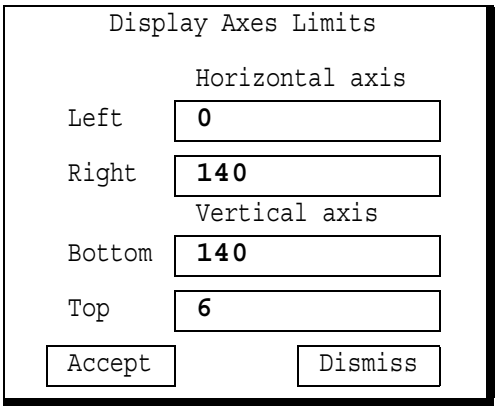
FILE ↓
Read file → Read model

Select *space.sp* from the list of files.



Use the **DISPLAY** menu to view the model:

DISPLAY ↓
Axes limits



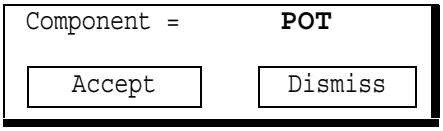
DISPLAY ↓
POST-processing

DISPLAY ↓
Refresh

The potential distribution can be seen by displaying contours with the component set to POT.

FIELDS ↓

Component

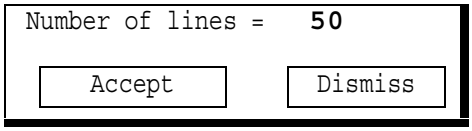


FIELDS ↓

Contour Plot → Style → Filled zones

Contour Plot → Style → Return

Contour Plot → Number of lines



Contour Plot → No refresh (*toggles to refresh*)

Contour Plot → Execute

The results are shown in Figure 12.3.

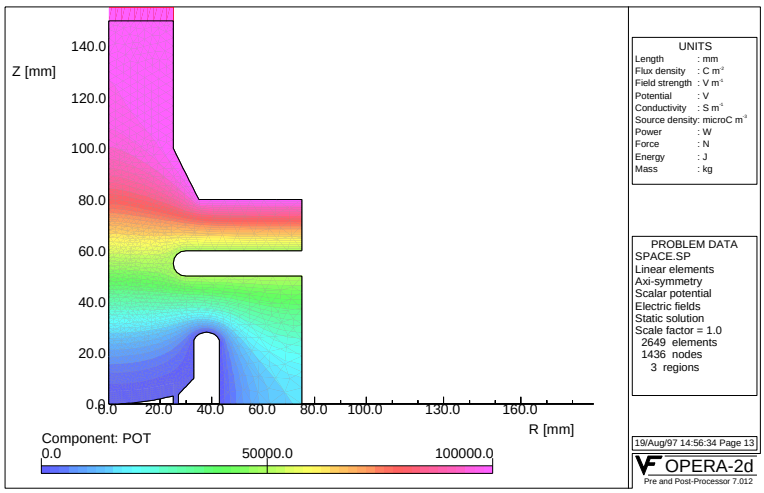


Figure 12.3 Electric potential distribution displayed on the model

There are several ways of displaying the tracks calculated by the space charge analysis program. This includes a display of the tracks over the geometry, while another is to display a three angle projection.

To display the tracks over the geometry, first select the file containing the tracks and then display the tracks. First refresh the display

DISPLAY ↓
Refresh

To select the file containing the tracks.

FIELDS ↓
Trajectories → Display → Select Track file

Choose the file *space_1.tracks*

To display the track file on the current geometry:

FIELDS ↓
Trajectories → Display → Display trajectories

giving the results shown in Figure 12.4.

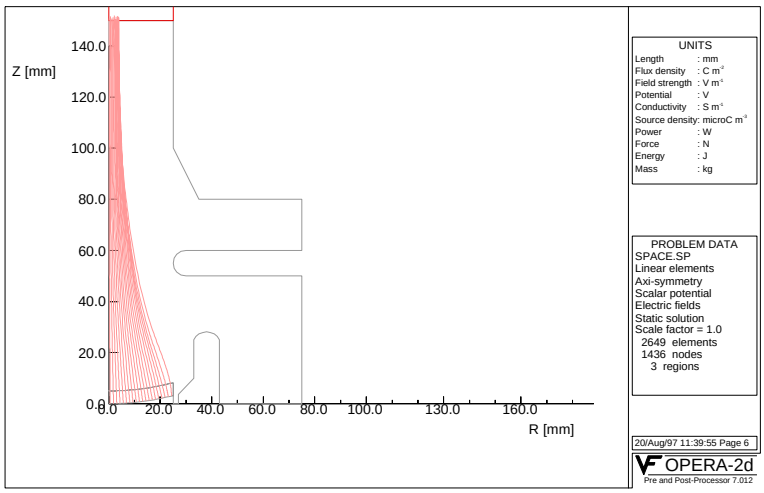
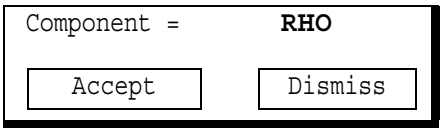


Figure 12.4 Space charge tracks displayed on the model

The charge distribution can be seen by plotting contours with the component set to RHO.

Component



Contour Plot → Execute

The results are shown in Figure 12.5.

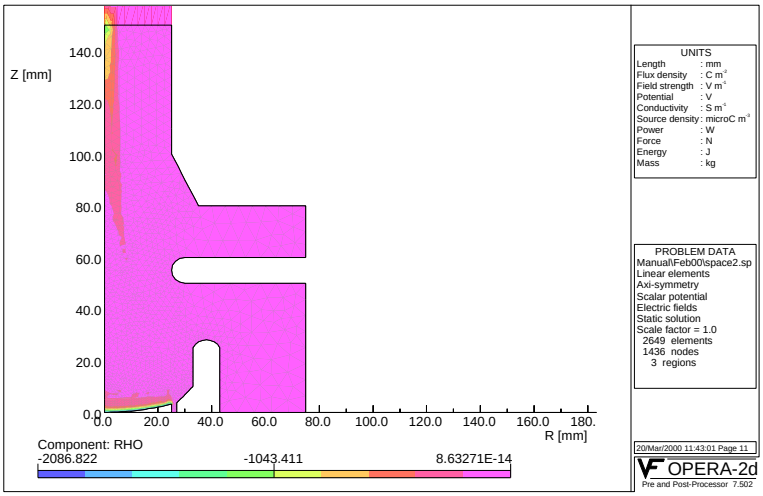


Figure 12.5 Charge Density displayed on the model

This concludes the Space Charge Example.

INDEX

Numerics

2D 4-5, 7-1, 8-1
3D 4-5

-A-

ac analysis 4-34, 5-18, 11-12
accumulate 8-35
accuracy 3-2, 4-5, 5-6, 8-35, 9-3
air 4-5, 4-18, 4-26, 7-3, 7-12, 7-16, 8-15
airgap 9-37
analysis 3-7
  ac 5-18
  space charge 5-54
  statics 5-15
  stress 5-74
  thermal 5-70
  transient 5-22
  velocity 5-28
analysis modules 3-1, 3-2, 4-32, 4-34, 7-22, 8-29, ... 9-44
arc construction lines 9-6, 9-22
aspect ratio 7-2, 8-2, 8-3, 8-28, 10-6
at intersection 8-7, 9-7
at old point 4-16, 7-8, 9-38
automatic mesh generator 8-27
axes 4-2, 4-3, 4-5, 7-2, 7-24
axi symmetry ...3-1, 3-5, 4-5, 7-25, 8-1, 8-4
  recommended solution potential in ST ... 5-16

-B-

background fields 5-20
background region ...4-18, 4-19, 4-23, 4-26, 7-16, ...8-15, 8-19, 8-20
back-space key 4-6
BH 7-15, 8-25, 9-39
  library 7-16
  tables 3-6
BH curve 4-32
  default 4-32
BH files
  Unix 2-4
  Windows 2-15
bias ...4-8, 4-9, 4-21, 4-23, 4-24, 4-29, 8-18
BMOD 4-39, 8-33
body force 5-77, 6-24, 10-24
boundary conditions 5-2, 5-4, 5-10
  `B normal = 0` ...4-31, 7-19, 8-26, 9-41
  far-field 4-31
  fixed temperature 11-6
  heat transfer 11-6
  load 10-15
  periodic symmetry 9-16, 9-41
  position constraint 6-23, 10-16, 10-23
  setting 4-31
  steady-state ac 5-20
  thermal 11-6, 11-13
  transient 5-26, 5-37, 5-45
BZ 4-41

-C-

C region 8-21
calculator 9-49
capacitance 6-5
cartesian coordinates 3-1
centimetres 4-2
change regions 4-29, 8-21, 9-36
charge density 12-14
circumferential stress 10-4
clear and reset 7-28
close polygon ...4-9, 4-11, 4-14, 4-18, 4-23, 7-7, ... 8-8
coercive force 5-13
coil 4-4, 4-5, 4-12, 8-14
coil former 4-4, 4-14, 4-23, 4-26, 4-28, 4-29
COMI Files 2-4, 2-15
command files 2-4, 2-15
  command input 5-79
command menu 4-2
complex permeability 5-20
component 4-39, 4-41, 7-26
conductor 4-12, 4-15, 4-16, 8-14, 9-19
conductor number 5-22, 5-33, 5-42
constants 9-49
constraints 10-8
construction lines 8-4, 8-7, 9-4, 9-7, 9-22
continuity 4-23, 5-4, 7-20, 8-19, 9-38
contour 7-24, 9-46
  plot 4-37, 4-39, 4-40, 10-17
coolant temperature 11-13
copying regions 9-16
coupled problems 10-5, 10-20, 11-9
CPU priority
  Windows 2-10
creating tables 6-24, 10-24, 11-13
CS region 8-21
current 7-10
current density .4-13, 7-10, 7-15, 8-3, 8-14, 9-21
cursor 4-2, 4-4, 4-16, 4-26
curvature 4-8, 4-9, 9-8, 9-25, 9-27
cylindrical polar coordinate 4-5

-D-

deformed plot 10-17
diffusion equation 3-1
disc 4-9, 4-12
DISPLAY 4-3, 4-25, 4-37
display options 8-30, 9-45, 10-17
dongle installation 2-8
down-arrow 4-7
draw background 4-19, 4-22
draw polygon ..... 4-7, 4-9, 4-10, 4-11, 4-14, 4-16, ...4-19, 7-3, 9-6, 9-22
drawing regions ... 4-6, 4-7, 4-9, 4-11, 4-12, 4-14, ... 4-18, 7-3, 8-7, 9-6, 9-22
driving functions 5-22, 5-33, 5-42
DXF files 5-79

-E-

E shaped core 7-3, 7-7
eddy current conductors ...5-19, 5-24, 5-35, 5-43
eddy currents 5-13, 7-22
electric 4-5
electrodes 12-1
electromagnetic 3-1
electron gun 12-1
electrostatics 3-1, 4-34, 5-13
elements 3-1, 4-5, 4-29, 8-21, 9-1, 9-36
emitter data 12-11
emitter direction 12-11
emitter files 12-2, 12-11
emitters 12-2
end effects 7-1
environment variables
  Unix 2-1
ERROR 5-6
error 4-34, 7-25
escape key 4-3, 4-7
excitation current 4-34
external circuits 5-19, 5-25, 5-35, 5-44

extra conditions 6-23, 10-23
extra options 10-15, 11-5

-F-

far-field boundaries 4-31, 7-19, 8-15
field averaging 3-7
field emission 5-63
file box 4-37, 7-23, 9-40
file names
Unix 2-3
Windows 2-14
files
.dxf 5-79
.mesh 4-35, 7-21
.op2 ..... 4-35, 7-21
.sa 10-17
.st 4-36, 7-23
.th 11-15
reading solution data 4-37
writing model data 4-35, 7-21
filled zones 4-40, 7-25
finite element mesh ..... 4-9, 4-24, 7-20, 8-2, 8-19, ... 8-27, 9-36, 10-5, 10-14
finite element method 3-1, 5-2
fixed temperature 11-3
flux density 4-39, 7-25, 7-27, 8-33, 9-46
flux lines 4-38, 7-25, 8-31, 9-46
force 6-24, 8-35, 9-47, 10-24
Fowler Nordheim field 5-62

-G-

Galerkin 5-3
gap region 3-5, 5-31
getting started 4-1
graph 4-41, 7-27, 8-33
graphics screen 4-2
graphics variables
Unix 2-2
Green's theorem 5-4
group regions 7-12
groups 7-12

-H-

H region 4-29, 8-21
hard magnetic material 9-24
HEAT 11-15
heat flux 11-1, 11-6, 11-13
heat transfer coefficient ... 11-2, 11-6, 11-13
Helmholtz equation 3-1
hysteresis 5-20

-I-

inductance 6-5
installation notes 4-1
insulator 11-2
integration 8-35, 9-47
integration surface 8-35
internal boundaries 7-20
isotropic material 4-32

-K-

Kelvin transformation 5-12

-L-

laminated 7-22
least squares 5-3
licenses 2-9
line contours 4-37, 7-24, 8-31, 9-46
line graph 7-27
line integration 5-7, 8-35, 9-47
linear elements 9-3, 10-6
loading BH data 8-26, 9-40
log files
Unix 2-4

-M-

magnetic materials 7-12, 7-15, 8-2, 8-25
magnetic vector potential 9-41
magnetisation direction 9-26
magnetostatics 3-1, 4-34, 5-13
material label 4-7, 7-12, 8-2, 8-7, 8-25, 9-19
material properties .....4-5, 4-6, 4-13, 4-14, 4-19, ...7-12, 8-7, 10-7, 11-3

stress analysis 5-76
thermal analysis 5-71
material type 4-6, 4-19
Maxwell stress 6-24, 8-35, 10-24
Maxwell's equations 5-13
mechanical data ... 6-22, 10-7, 10-14, 10-23
menus 4-2, 4-3
mesh 4-23, 4-24, 4-31, 4-34, 9-15
mesh errors 9-43
mesh file 4-35
mesh generator .4-5, 4-24, 4-29, 4-32, 7-20, 9-42
message box 4-5
modified rA 4-5, 4-37, 7-25, 8-4, 8-31
modified vector potential 8-31
modify regions .4-26, 4-29, 7-13, 8-23, 9-8, 9-18, ... 9-25
modify sides 4-26, 4-29, 8-23, 9-8
motor 9-1
mouse 4-2, 4-16, 4-26, 7-9, 8-7
mouse input 4-16, 7-8, 8-7, 9-7
mu 4-7
multiple turns 4-12

-N-

neodymium iron boron 9-40
Newton-Raphson method 5-5
non-linear 3-2, 4-34
non-magnetic 7-12, 8-14, 9-25

-O-

open boundaries 5-11
OPERA Console 2-9
opera2.comi 2-4, 2-15
opera2_de.comi 2-4, 2-15
OPERA-2d 3-1
OPERA-2d/RM
rotor speed 5-37
Opera2d_PP_n.log
Unix 2-4
Windows 2-14
Opera2d_PP_n.lp
Unix 2-4
Windows 2-14

-P-

parameter box 4-6, 4-8
particle tracking 12-15
particle trajectories 12-15
Peclet number 5-28
perfect insulator 11-2
periodic symmetry 9-41
permanent magnet ...4-34, 5-26, 5-37, 5-45, 9-1, ...9-24
permanent magnet motor 9-1
permeability 4-7, 4-13, 4-14, 5-20, 8-7
perodicity conditions 5-12
perturbation fields 5-20
phase 9-26
plane strain 10-2
plane stress 10-2
plasma free surface 5-63
point 4-7, 4-9, 4-15, 4-16, 7-10
Poisson equation 3-1, 5-2, 5-70
Poissons ratio 6-23, 10-3, 10-15, 10-23
polygon 4-5, 4-7, 4-9, 4-18
post processing 3-7
POT 4-39, 7-25
pre processor 3-5, 4-1, 5-1, 8-3, 9-3
pressure 10-9
primary winding 7-1, 7-10
program size
Unix 2-5

-Q-

Q region 8-21, 9-37
quadratic elements 7-25, 9-3, 10-6
quadrilateral regions ...4-29, 8-2, 8-15, 8-21

-R-

read model 4-37, 7-23, 11-7
refinement 4-34

refresh 4-3, 4-25, 4-37
region 3-5, 4-5, 4-12, 7-3
region shape code 9-36
relative permeability 4-7
relaxation factor 5-5
remesh 7-23
replication 9-16, 9-18, 9-21
residual 5-5
rho 12-14
Richardson-Dushman 5-55
RMS error 7-25, 8-31, 9-3
RMSERROR 5-6
rotating machine analysis 4-34
rotational periodicity 9-1
rotational symmetry 9-18
rotational velocity 5-29
rotor 9-21

-S-

saving BH data 8-25
saving model data 8-28, 11-7
Schottky field emission 5-62
secondary winding 7-1, 7-10
select material 4-40
shape code H 12-8
shape function 5-3
shear modulus 10-1
shear stress 10-2
SI units 4-2
side 4-9, 4-12, 4-16
skin depth 5-7
solenoid 8-1
solution file 4-36, 7-22, 8-29
solution potential 4-5
solution symmetry 4-5
Space charge 12-1
Space charge analysis module 12-2
space charge beam analysis 5-54
static adaptive analysis 4-35
static analysis . 4-34, 4-35, 5-15, 7-22, 8-29, 10-22
theory 5-74
stator 9-7
steady-state ac 4-34
analysis 5-18
steel 7-3
step-down 7-1
stiffness matrix 5-4
strain 10-3
stray fields 8-17
stress 10-1
stress analysis 3-8, 4-35, 10-1, 10-20
loads and constraints 5-75, 5-76
material properties 5-76
post processing 5-78
pre processing 5-76
stress component 10-18
stress failure 10-10
subdivision ..... 4-9, 4-24, 4-27, 4-29, 8-18, 8-22, ...9-8, 9-20, 9-27, 9-38
swap space
Unix 2-5
symmetry 9-1
system icon tray 2-9
system manager
Unix 2-2

-T-

tables 12-14
stress analysis 5-77
TEMP 11-7, 11-8, 11-16
temperature 11-1, 11-3
text input 4-21
text window
Windows 2-13
thermal analysis 3-8, 4-34, 4-35, 11-1
boundary conditions 5-70
heat sources 5-72
material properties 5-71
post processing 5-72
theory 5-70
thermal conductivity 11-1, 11-5, 11-13

thermal expansion integral 10-7, 10-9
thermal material 11-5, 11-12
thermal saturation 5-55
time stepping 4-34, 5-25, 5-36, 5-44
time tables 5-23, 5-34, 5-43
torque 9-47
transformer 7-1
transient analysis 4-34, 5-22

-U-

unfinished polygon 4-9
UNITS 4-2
units 4-2, 6-24, 7-2, 8-3, 9-3, 10-24
Unix implementation 2-1
up-arrow 4-7
updating licenses 2-9
upwinding 5-29
user defined variables 9-49

-V-

variables 9-49
variational methods 5-3
vector potential 7-19, 7-24, 8-32
velocity analysis 4-34, 5-28
vfdir
Unix 2-1
VFGRAPHICS
Unix 2-2
VFINV
Unix 2-2
VFWINDOWH
Unix 2-2
VFWINDOWW
Unix 2-2
virtual memory
Unix 2-5

-W-

weighted residuals 5-3
winding 7-10, 7-15
write model 8-28, 9-43, 10-16, 11-7

-X-

X move 4-10, 4-14, 4-17, 7-4
XY input ... 4-7, 4-9, 4-13, 4-15, 4-16, 4-20, 7-3
XY symmetry 4-37, 7-24, 8-4

-Y-

Y move 4-10, 4-13, 4-18, 7-4
Youngs modulus ... 6-23, 10-1, 10-15, 10-23

-Z-

zone contours . 4-39, 4-40, 7-25, 8-33, 9-46
zoom display 4-21, 8-24, 9-39